



---

**SISTEMA DE CONVERSIÓ D'ÀUDIO ENVOLVENT 5.1 A ÀUDIO BINAURAL ESPACIAL**

**Jaume Grande Cuello**

---



**UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH**

---

**Escola d'Enginyeria de Terrassa**

**JUNY DE 2015**

**TUTOR:**

**Andreu Balastegui**



# INDEX

0- Agraïments .....	4
1- Introducció .....	5
2- Estat de l'art.....	7
2.1- Acústica.....	7
2.1.1- Fonaments .....	7
2.1.2- Conceptes bàsics .....	8
2.2- Fisiologia de l'oïda .....	12
2.3- Localització d'una font sonora a l'espai.....	13
2.4- HRTF .....	16
2.4.1- HRTF Què és .....	17
2.4.3- Models.....	19
2.4.3.1- Aproximació teòrica, model analític: .....	19
2.4.3.2- Aproximació empírica .....	20
2.4.3.3- Bases de dades, matriu de respostes impulsional	21
2.4.3.4- Mesures.....	22
2.5- ffmpeg.....	25
2.6- Contenedors multimèdia .....	26
2.7- Formats de vídeo .....	28
2.8- Formats d'àudio .....	29
2.9- So envoltent 5.1 .....	29
3- Implementació.....	32
3.1- Matlab, per què? .....	32
3.2- Esquema general del programa.....	33
3.3- 1ª Fase: extracció de l'àudio .....	33

3.4- 2ªFase: Filtratge .....	35
3.5- Retard interaural .....	38
3.6- 3ª Fase: Unir vídeo i àudio.....	41
3.6- GUI – Graphical User Interface .....	42
4 – Estudi d'optimització de la posició virtual dels altaveus .....	45
4.1- Altaveus frontals.....	46
4.2- Altaveus trasers .....	47
4.3- Anàlisi subjectiu dels resultats .....	48
5 – Conclusions .....	51
6- Bibliografia.....	53
7- Annexes .....	54
7.1- "gui.m" .....	54
7.2- "filtr_HRTF.m" .....	69
7.3- "distancia_LR.m" .....	71
7.4- "Delay_Samples.m" .....	73
7.5- "calcul_N_samples.m" .....	74
7.6- "recerca.m" .....	76
7.7- "scrip_analisis.m" .....	77

## 0- Agraïments

Probablement oblidi algú injustament o els citi en ordre equivocat... Vull començar per la meva dona a qui tants mals de caps li he donat en el camí fins aquí, per ser la meva fortalesa i el millor mirall en qui puc reflectir-m'hi. A la meva petita Yulia per ser la gran motivació i font de forces en aquests últims i durs anys.

Aquests pares a qui els ha tocat un fill com jo, mai s'ho van imaginar, ni tan sols somiar que algun dia podria arribaria a assolir aquestes fites, part de l'èxit és d'ells i als valors amb què van educar-m'hi, mil gràcies per tot el que mai us podré tornar.

Per descomptat al meu tutor Andreu Balastegui, tot un plaer i grata sorpresa. Gràcies per l'ajuda quan no he sabut cap on anar, gràcies pels ànims quan els he necessitat i sobretot gràcies per la paciència i comprensió.

Gràcies també a tu, que vas encoratjar-m'hi tant a estudiar i vas tindre més fe en mi que jo mateix, encara me'n recordo de tu. Ningú és mort mentre algú encara el pugi recordar, i jo no ho faré.

I per acabar no puc oblidar-me de totes aquelles persones que van pensar que mai podria arribar fins aquí, GRÀCIES, en la decepció vaig trobar sempre la motivació.

# 1- Introducció

El món audiovisual evoluciona constantment, sempre en busca de major definició, espectacularitat i realisme. El so al llarg del temps ha anat evolucionant però en menor mida que el vídeo, del mono a l'estèreo fins arribar a la gran revolució del 5.1, el cinema a casa. El 5.1 va néixer amb la idea de so envoltent. Entenem per sistema d'àudio envoltent 5.1 com un conjunt de cinc altaveus situats al voltant d'un individu que permeten oferir un so espacial, és a dir, localitzat a l'espai, com els d'un equip "*home cinema*", però sempre amb una disposició molt específica tant dels altaveus com del oient. La gran pega dels sistemes de so convencionals és la nul·la portabilitat d'aquests mateixos.

Els humans tenim la capacitat de localitzar i estimar la posició d'una font de so a partir de l'anàlisi de les ones que ens arriben a només dos punts (les oïdes). Les ones provinents d'una determinada font sonora es reben a diferents instants de temps i amb diferents intensitats a cada oïda. En la localització de fonts a l'espai, a part de la mesura de la intensitat i la diferència de l'interval de temps entre les dues oïdes, també hi intervenen la forma del pavelló auditiu, el tors i el cap.

Aquest projecte pretén desenvolupar un programa per a convertir aquest so en binaural, per tant que es pugui reproduir en uns auriculars mantenint l'efecte tridimensional. El programa desenvolupat no afectarà la gravació del so, simplement entrarà en joc en el procés de reproducció, mitjançant l'aplicació d'uns filtres anomenats Head-Related Transfer Functions (HRTF) a cada un dels 6 canals de l'àudio 5.1. Les HRTF són unes funcions de transferència que modelen el procés de localització binaural que realitzem els humans de manera natural. Un cop implementats aquests filtres, el "*software*" creat permetrà reproduir sons provinents de qualsevol dels cinc punts estudiats. L'efecte obtingut serà el mateix que si s'escoltés la mateixa gravació en una sala equipada amb un equip "*home cinema*".

Aquest tipus de desenvolupaments obre un món de possibilitats al món audiovisual tant en àmbits tan evidents com el cinema però també en d'altres com videojocs, aplicacions mòbils o una temàtica a l'alça actualment com la realitat virtual. Seria espectacular poder estar immers, no sols visualment sinó, auditivament en allò que estàs veient, poder col·locar uns auriculars a una entrada de 3.5 mm al cinema i poder sentir que estàs immers dintre de la pel·lícula, localitzar amb l'oïda d'on venen els sons. La nostra idea és la de facilitar a l'usuari la possibilitat de poder portar a sobre i allà on vulgui el so multicanal sense limitacions d'espai ni disposicions, sols amb uns auriculars convencionals estèreo.

Entendre i desenvolupar el "software" és important també per a l'estalvi de recursos que pot oferir en nombroses aplicacions audiovisuals. A més, servirà de punt de partida per a futures investigacions per estendre aquesta localització del so a tots els punts possibles de l'espai. Aquesta extensió permetrà implementar el resultat en situacions molt diverses i que poden ser molt interessants com són el guiatge per a gent invident, l'audioguia per a museus i la localització de sons en cabines de pilot.

A les properes pàgines d'aquesta memòria es podrà trobar la documentació suficient per entendre el desenvolupament del projecte; incloent des de la descripció del Sistema Auditiu Humà (l'eina fonamental), les capacitats per part dels humans en la localització de elements audibles a l'espai, les HRTF (què són, com es troben i com funcionen), breus explicacions bàsiques i definicions acústiques i la seva implementació en Matlab com a eina de desenvolupament del "software" que haurà de separar els canal 5.1 i aplicar les HRTFs a cada canal per tal d'obtenir una nova mescla estèreo amb l'àudio binaural que contindrà la informació espacial.

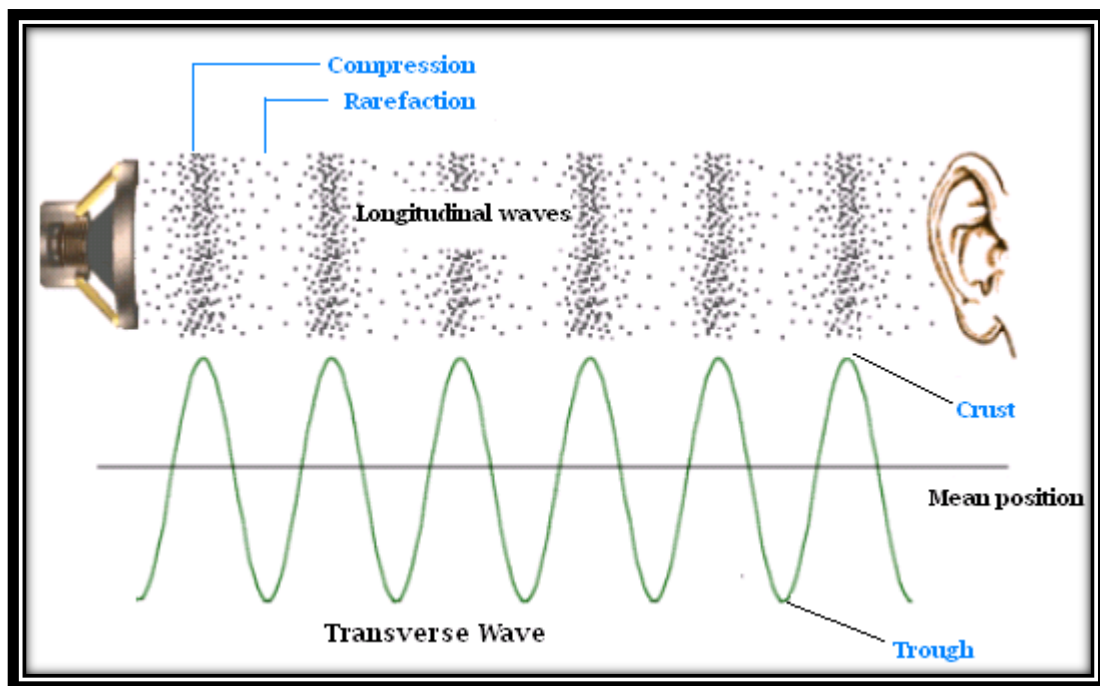
## 2- Estat de l'art

### 2.1- Acústica

L'acústica és la ciència que estudia el so, tant com es produeix, com és transmet i els seus efectes.

#### 2.1.1- Fonaments

El so és una ona mecànica, una pertorbació transmesa a través d'un medi elàstic.



*Figura 1. Esquema de la producció de una ona sonora*

Podem considerar el so en la seva doble vessant:

**Físic:** Vibració mecànica que es desplaça per un medi elàstic i que és capaç de produir una sensació auditiva.

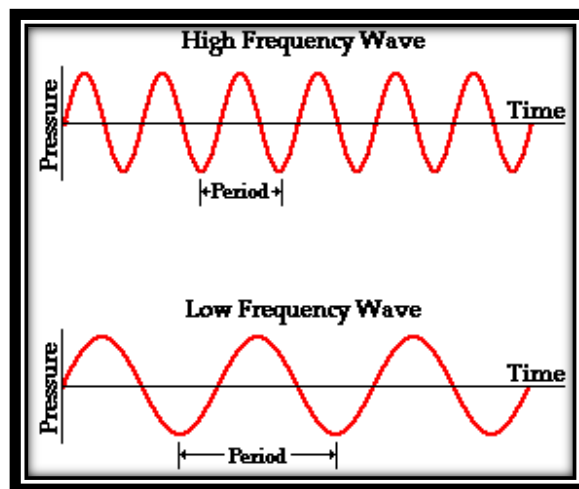
**Fisiològic:** Sensació auditiva produïda per una vibració mecànica que ha utilitzat un medi elàstic per propagar-se.



## 2.1.2- Conceptes bàsics

### **Freqüència:**

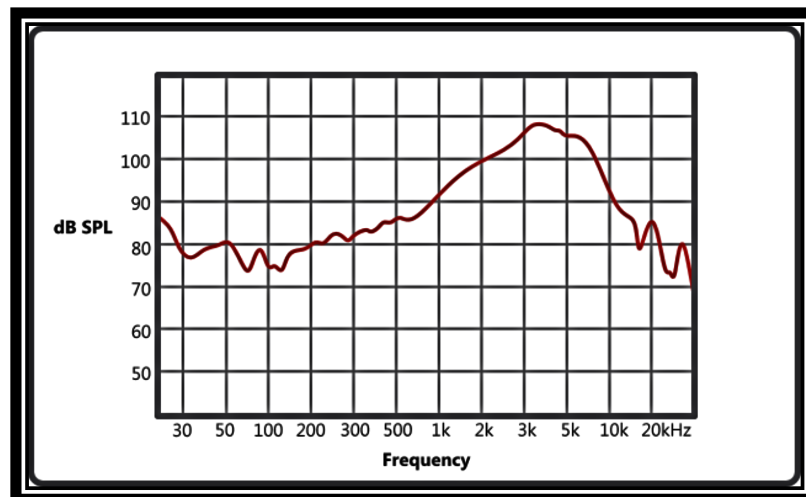
És el número de cicles complets (períodes) o vibracions per segon. La unitat de freqüència és l'Hertz (Hz). A les altes freqüències trobem els sons aguts, mentre que a les baixes freqüències trobem els sons greus.



*Figura 2. Gràfica de la freqüència*

### **Espectre freqüencial:**

L'espectre de freqüència d'un fenomen ondulatori (sonor, lluminós o electromagnètic), superposició d'ones de diverses freqüències, és una mesura de la distribució d'amplituds de cada freqüència. També es diu espectre de freqüència al gràfic d'intensitat enfront de freqüència d'una ona particular. L'espectre audible de l'ésser humà va des dels 20 Hz fins als 20 KHz i la sensibilitat de l'oïda humana no és la mateixa per tot el rang freqüencial audible.



**Figura 3. Espectre freqüencial**

L'espectre audible es pot classificar en funció dels tons (atribut que qualifica un so com més greu o més agut

- Tons greus:** Freqüències baixes, corresponents a les 4 primeres octaves\*, des dels 20 Hz fins els 250 Hz.
- Tons mitjos:** Freqüències mitges, corresponents a les octaves cinquena, sexta i setima, des de 256 Hz fins 2 KHz.
- Tons aguts:** Freqüències altes corresponents a les últimes tres octaves, des de 2 KHz en endavant.

\*Octava: És l'interval de freqüències comprés entre una determinada freqüència i una altre igual al doble de l'anterior.

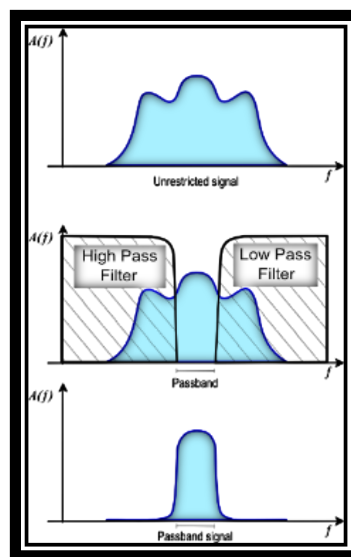
### Transformada de Fourier:

La transformada de Fourier descompon una funció temporal (un senyal) en les freqüències que la constitueixen. Aquesta descomposició resultant és una funció complexa, el valor absolut de la qual representa la quantitat de cada freqüència present en la funció original, i l'argument complex de la qual és el desfasament de la sinusoide bàsica en aquella freqüència. Si bé l'aplicació de la transformada de Fourier no es limita només a funcions temporals, el domini de la funció original se sol anomenar domini temporal. La transformada és anomenada domini freqüencial.

El terme transformada de Fourier fa referència tant a la representació en el domini freqüencial com a l'operació matemàtica que associa el domini freqüencial a una funció temporal.

### Filtratge:

És part del processament del senyal. S'utilitza per adequar (atenuar o amplificar) alguna part de l'espectre freqüencial i d'aquesta manera podem modelar l'espectre d'un senyal per tal de poder eliminar o aïllar parts de l'espectre que ens poden interessar.



*Figura 4. Imatge d'un filtre passa-banda*

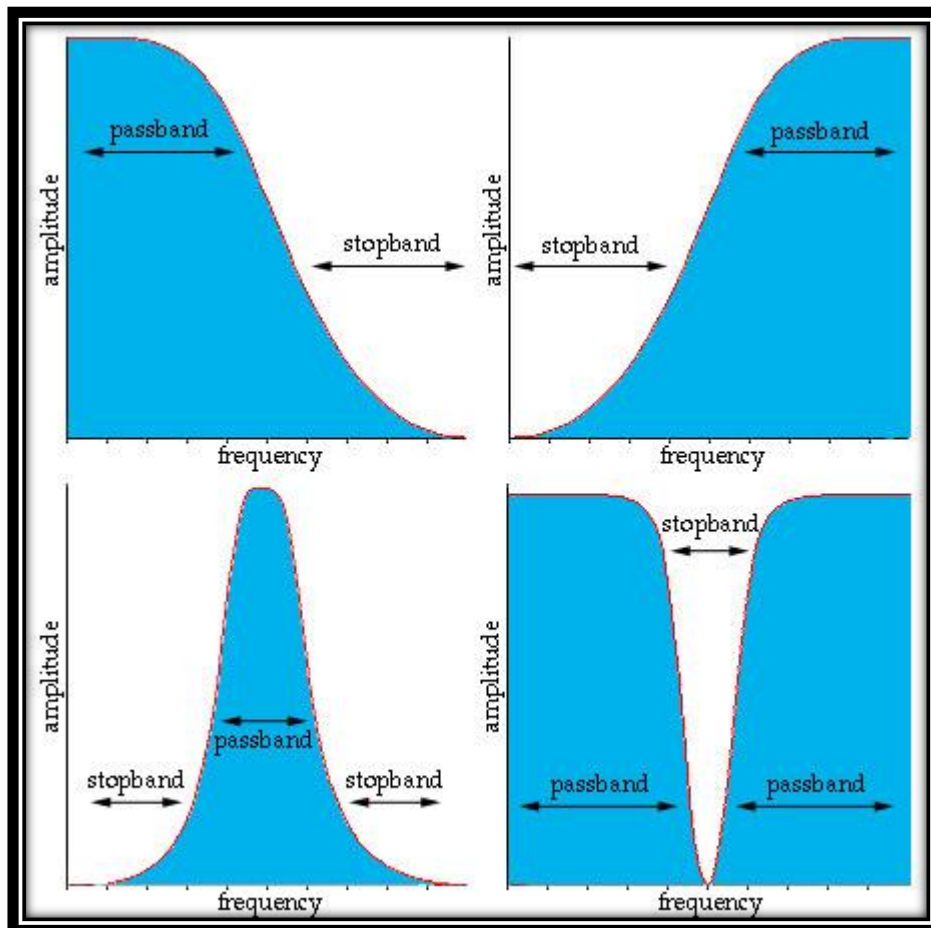
Segons com actuen, els filtres poden ser:

- **Conformadors de freqüència:** Varïen l'espectre.
- **Selectius:** Poden eliminar part de l'espectre i deixar passar un altra part.

De filtres existeixen tant digitals, com analògics. Analògics hi ha de dos tipus, els passius que no precisen alimentació i els actius que si en necessiten.

Segons la seva funció els filtres es classifiquen:

- **Passa-baixos:** Són aquell tipus de filtre que deixa passar les baixes freqüències i elimina les altes.
- **Passa-alts:** Són aquell tipus de filtre que deixa passar les altes freqüències i elimina les baixes.
- **Passa-banda:** Són un tipus de filtre resultant, de la combinació d'un filtre passa-baixos i un passa-alts que deixa passar una determinada banda freqüencial i n'elimina la resta.
- **Elimina-Banda:** És l'invers de l'anterior, és un filtre resultant, de la combinació d'un filtre passa-baixos i un passa-alts que elimina una determinada banda freqüencial i deixa passar la resta.

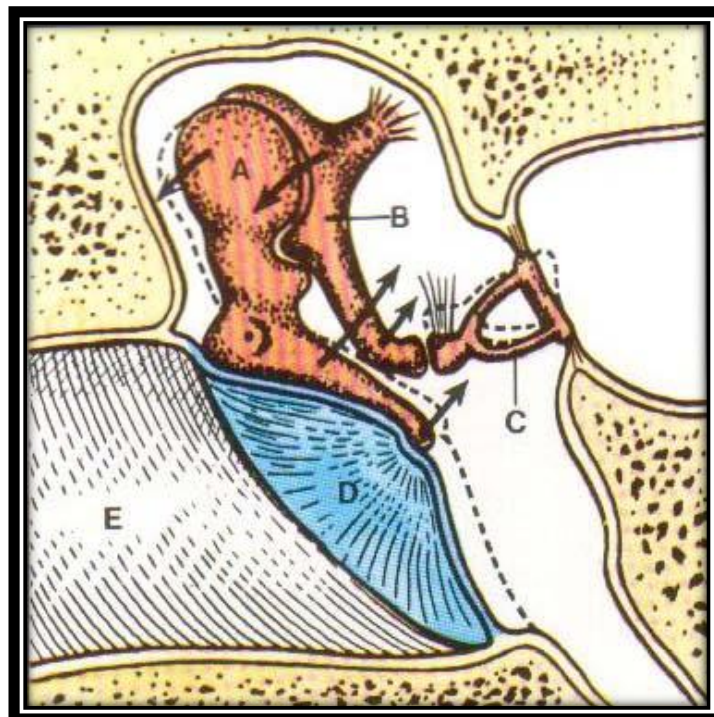


*Figura 5. D'esquerra a dreta i d'amunt avall: Exemple de filtre passa-baixos, filtre passa-alts, filtre passa-banda i filtre elimina-banda*

## 2.2- Fisiologia de l'oïda

L'oïda és un conjunt d'òrgans que tenen la principal funció de dotar d'equilibri i de captar el so de l'entorn que ens envolta.

El so entra a l'oïda pel canal auditiu extern i fa que la membrana del timpà vibri. Les vibracions transmeten el so en forma d'energia mecànica, mitjançant l'acció de palanca dels ossos cap a la finestra oval. Després, aquesta energia mecànica es transmesa pels líquids de l'oïda a la còclea, on es converteix en energia elèctrica que viatja pel nervi vestibul-coclear cap al sistema nerviós central, on es analitzat i interpretat com so en la seva forma final. Durant aquest procés de transmissió, les ones sonores troben protuberàncies cada cop més petites, des del pavelló auricular fins la petita finestra oval, que resulten en increment de l'amplitud (o volum) del so. Les ones sonores transmeses per la membrana del timpà als ossets de l'oïda arriben al cargol, que és l'òrgan encarregat de l'audició situat al laberint o oïda interna.



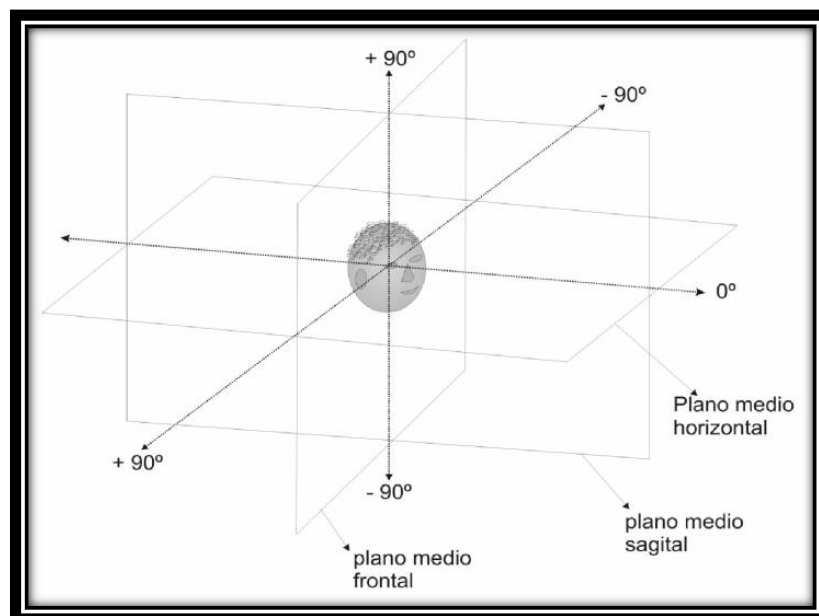
*Figura 6. Esquema de la transmissió del so en l'òrgan auditiu*

*A-Martell. B-Junc. C-Estrip. D-Membrana del timpà. E-Conducte auditiu extern. Les fletxes indiquen el sentit en que es desplacen els ossos*

Un osset important és l'estrep, que balanceja i estableix les vibracions (ones) en els líquids continguts en el laberint. Aquestes ones líquides, al mateix temps, causen el moviment de la membrana basilar que estimula a les cèl·lules de l'òrgan de Cort per moure's en forma de ones. Els moviments de la membrana estableixen les corrents elèctrica que estimula les diverses àrees de la còclea. Les cèl·lules ciliades inicien un impuls nerviós que es codifica i transfereix a l'escorça auditiva del cervell, on es descodifica en la forma d'un missatge sonor.

## **2.3- Localització d'una font sonora a l'espai**

La localització sonora és la percepció de la posició d'una font sonora en el pla horitzontal o azimuth (esquerra-dreta), en el vertical o elevació (amunt - avall) i la percepció de la distància relativa entre receptor i font.



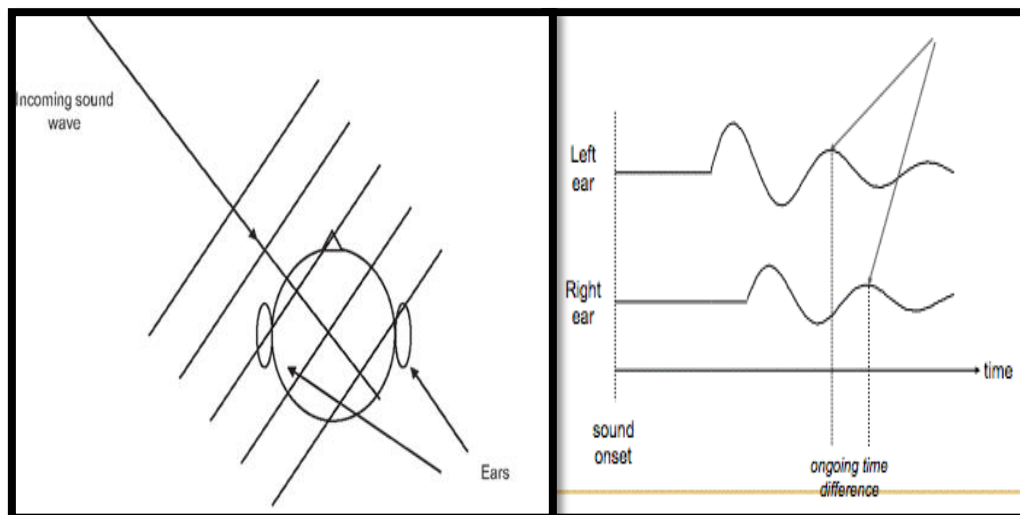
**Figura 7. Representació esquemàtica de plans espacials. Una font sonora a 0° d'azimut i 0° d'elevació està situada just davant del receptor a l'altura de les seves oïdes**

Es denomina pla mig horitzontal el qual passa per les oïdes i divideix imaginàriament al cos humà en dues meitats, superior i inferior. El pla vertical el divideix en dreta i

esquerra, el pla vertical mig sagital, on no existeixen diferències binaurals, el separa en dues meitats iguals. El pla frontal és el que passa per les oïdes a 90° del pla mig horitzontal i divideix al cos en part devantera i darrere. Aquests plans defineixen a l'espai els camps en dret (simbolitzat amb +), esquerre (simbolitzat amb -), superior (simbolitzat amb +) i inferior (simbolitzat amb -).

Els principals factors que intervenen perquè una persona pugui localitzar sons en el pla horitzontal a l'espai en un camp lliure són:

- **La diferència interaural de temps (ITD):** És la diferència temporal amb què un front sonor arriba a l'oïda esquerra i a l'oïda dreta (per les seves sigles en anglès, Interaural Time Difference).
- **La diferència interaural en el nivell sonor (ILD):** Es defineix com la diferència d'amplitud entre les dues oïdes quan el front d'ona arriba a l'oïda esquerra i a l'oïda dreta (per les seves sigles en anglès, Interaural Level Difference).

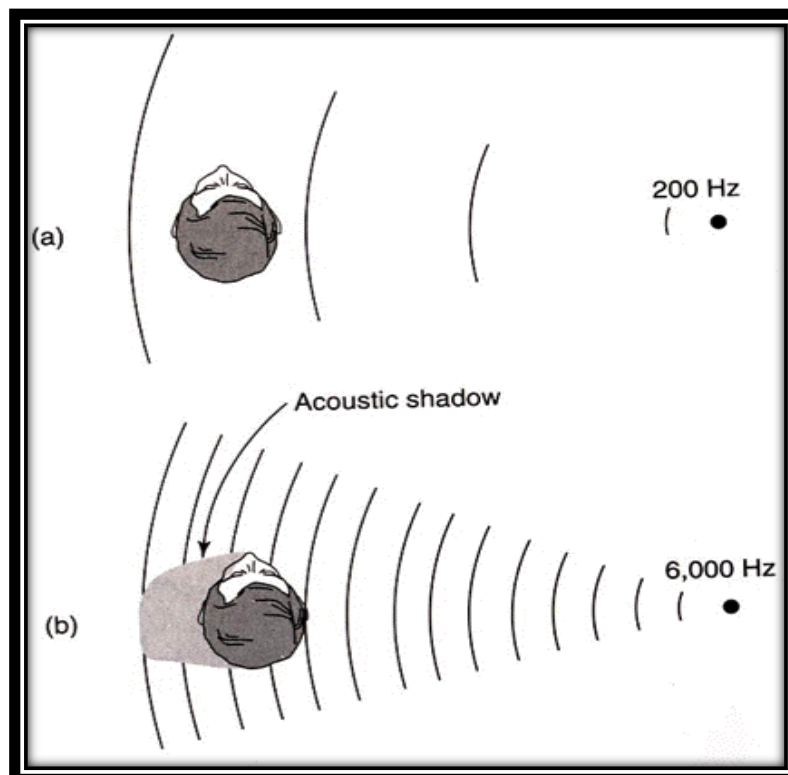


**Figura 8. Afectació de la temporal del retard interaural**

Ambdós factors són importants en la percepció dels sons originats en el pla horitzontals, podríem dir que les claus per la percepció espacial és la diferència de fase i amplitud que presenta un mateix senyal arribant a cada una de les oïdes. Però vora els 1.5 KHz la longitud d'ona passa a ser semblant al diàmetre cranial, i la diferència interaural esdevé

confusa donat que la diferència de fase no és suficientment gran per poder ser apreciada.

Per freqüències per sota dels 1.5 KHz el cap comença a fer ombra acústica a l'oïda més allunyada de la font, el que significa que l'amplitud del senyal que arriba a l'oïda que està a l'ombra acústica és menor que l'altre i és un factor molt important per la percepció espacial per a freqüències per sota dels 1.5 KHz. A més, el cap es comporta com una barrera acústica, que deixa passar les baixes freqüències i reflecteix les altes freqüències.



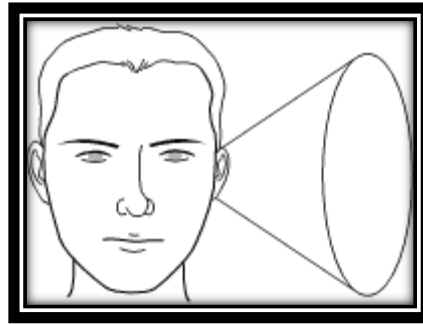
**Figura 9. Afectació de la freqüència a l'ombra acústica creada pel cap:**

**(a) A baixa freqüència, es pot apreciar com el cap no crea cap ombra acústica.**

**(b) Es veu com a altes freqüències el cap crea ombra acústica.**

La teoria de l'ITD i ILD expliquen la percepció horitzontal (azimut), però hi ha una gran ambigüitat per determinar quan una font ve des d'un punt més elevat o no, hi ha infinitat de punts diferents amb retards temporals i d'amplitud semblants, aquest conjunt de punts fou nomenat "con de confusió".





*Figura 10. Con de confusió*

També és conegut l'efecte de filtratge causat per la interacció del so amb el nostre propi cos com poden ser els plecs del pavelló de l'orella (pinnes), tors i espatlles. Aquest efecte de filtratge és molt important per poder distingir la procedència d'una font quan aquesta procedeix del pla vertical, en el qual la ITD i la ILD esdevenen zero en un cas de cap ideal.

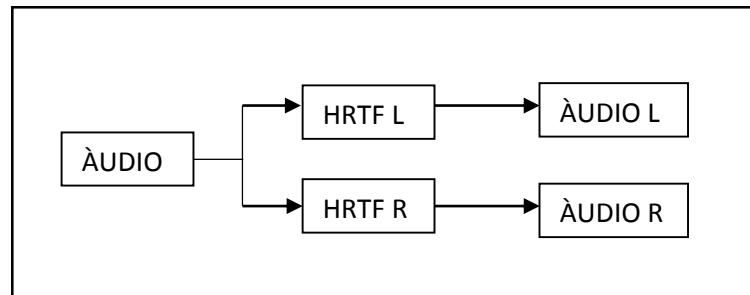
Per tant en unes altres paraules podríem dir que la posició percebuda de la font sonora en el pla horitzontal depèn tant del valor de ITD com de ILD mentre que en el pla vertical mig sagital, on com es recordarà no existeixen diferències interaurals, la posició percebuda de la font dependrà de l'efecte de filtratge causat pel nostre propi cos. La percepció de la distància es veu afectada, en canvi, per una constel·lació d'indicis d'intensitat, reverberacions i contingut espectral de l'objecte sonor.

## **2.4- HRTF**

La localització d'una font a l'espai és una qualitat molt important pels éssers humans. Les HRTF són unes funcions utilitzades en acústica les quals són el reflex freqüencial de com percep el so un individu en particular i amb les seves característiques físiques concretes. Cada individu tenim HRTFs diferents, a vegades semblants, amb les quals es pot caracteritzar l'espectre freqüencial d'un so tal qual arriba a la conca de l'oïda. Això ens permet dotar a l'espectre freqüencial d'un so qualssevol d'aquestes mateixes característiques, modificar-lo tal qual el rebria l'oïda en particular (dreta o esquerra) de l'individu al qual pertany la HRTF. Un so situat en l'espai és modelat amb un parell de

funcions de transferència, una per a cada oïda, l'efecte s'aconsegueix filtrant un senyal d'àudio per aconseguir l'efecte d'espialitat.

Podríem dir que el resultat és que si filtrem un àudio amb una HRTF és que si un oïdor amb uns auriculars escolta el mencionat àudio, sentirà el so captat com si l'estigués envoltant, envellant-lo en l'espai.



*Figura 11. Esquema de filtratge amb HRTF*

## 2.4.1- HRTF Què és

Existeixen diversos estudis que intenten obtenir la resposta freqüencial de l'escolta espacial. Veure com queda modificat l'espectre freqüencial, emes per una font situada a un punt de l'espai concret, en arribar al canal auditiu:

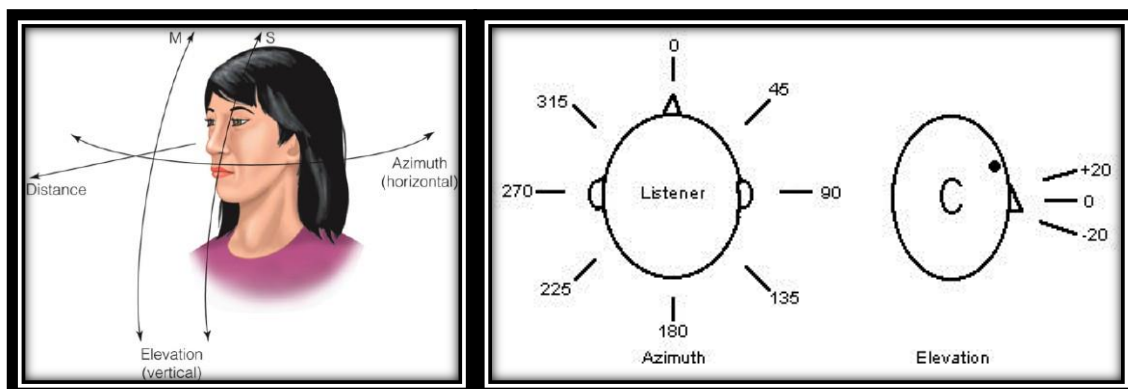
- Model Físic
- Mesures empíriques
- Simulacions computacionals (recerca més recents)

Una Head-related transfer functions (HRTF) es podria definir com la resposta freqüencial d'una oïda (dret o esquerra), mesurat amb una font sonora des d'un punt concret de l'espai a camp lliure al canal auditiu. Una HRTF captura les transformacions d'una ona de so propagant-se des de la font fins a les nostres oïdes, algunes de les transformacions inclouen difraccions i reflexions sobre parts del nostre propi cos tals com el nostre cap, pinnes, les espatlles i el tors. Com una conseqüència, amb aquestes funcions som

capaços de crear la il·lusió de localització del so a l'espai mitjançant uns auriculars en l'escolta.

Cal tenir en compte que les HRTF incorporen la informació tant de ITD com la de ILD. Els retards interaurals van intrínsecs en la HRTF com variacions de fase mentre que la informació d'ILD va directament relacionat amb la potència global del filtre.

Les HRTF són funcions que depenen de la freqüència, el que vol dir, que és la transformada de Fourier, en aquest cas, d'una head-related impulse response (HRIR) la qual està en funció del temps, no de la freqüència. Cal destacar que és una funció complexa definida una per cada orella. La HRTF també és altament dependent de la localització font sonora en relació amb l'oient, la qual cosa és la raó principal per la qual som capaços de localitzar una font sonora.



**Figura 12. Mostra gràfica d'azimut i elevació**

La resposta a l'impuls (HRIR), ha sigut la utilitzada en aquest projecte per filtrar cada canal d'àudio.

```
rightChannel=conv(audio,xr);
leftChannel=conv(audio,xl);
```

**Figura 13. Codi extret del projecte, de la funció *filtr\_HRTF.mxr* i *xl* son *xl* HRIR de cada oïda.**

Típicament les HRTF es mesuren a partir d'éssers humans o maniquins. Per ambdues oïdes, esquerra i dreta, en un radi fix des del cap de l'oient les HRTF són obtingudes a molt diversos azimuths (direccions d'esquerra a dreta) i elevacions (direccions d'amunt a avall).

## **2.4.3- Models**

És molt important la precisió a l'hora de l'adquisició de les HRTF, és el factor més important a l'hora de produir so en tres dimensions. Bàsicament hi ha dues maneres d'obtenir la informació de les HRTF, depenent de si l'aproximació es basa en un modelatge físic o si s'ha extret de mesures empíriques, distingim entre dues formes d'obtenir l'HRTF.

- Aproximació teòrica, modelat físic
- Aproximació empírica, matriu de respostes impulsional.

### **2.4.3.1- Aproximació teòrica, model analític:**

La HRTF es pot calcular utilitzant en la teoria de la difracció acústica, basat en la simplificació dels models humans com a models sense cap, cap esfèric, model del cap el·lipsoïdal i el mode del ninot de neu. Això simplifica molt els càlculs, donat que totes les irregularitats del cos humà, tals com la forma del cap o la forma del pavelló auditiu, són omeses.

Es realitzen aproximacions físiques amb càlculs matemàtics com per exemple utilitzar la distància entre les dues oïdes com a retards temporals, l'ombra acústica generada pel cap la modelen utilitzant un filtre passa baixos i l'efecte de l'orella externa com un filtre de pinta.

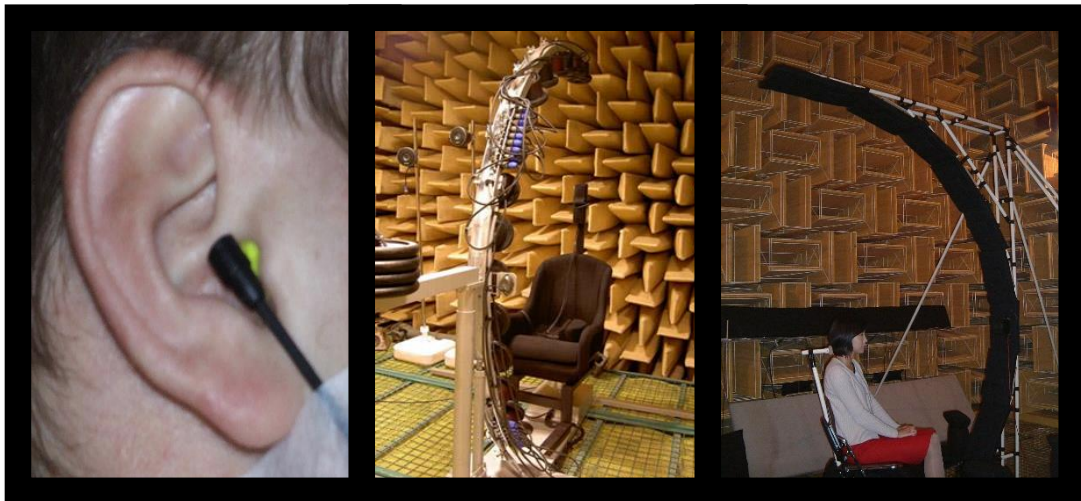


*Figura 14. Model del "ninot de neu" utilitzat en el model matemàtic*

La percepció subjectiva de les proves realitzades amb aquest tipus de funcions no és gaire satisfactòria comparades amb la realitat, però sí suficientment bones per a fer càlculs teòrics de retards o de nivells sonors interaurals.

### **2.4.3.2- Aproximació empírica**

Aproximació basada en l'obtenció de respostes impulsional empíricament en laboratoris amb sales acústicament preparades (normalment sales anecoiques). Ja bé sigui amb un ninot o amb un humà el procés és el mateix. Es col·loquen uns micròfons d'alta precisió i molt petits un en cada oïda, després s'estimulen amb un soroll amb espectre conegut (blanc o rosa) radiats per uns altaveus d'espectre el més pla possible. L'espectre freqüencial del so original i el del captat són ben coneguts, per tant si els comparem entre ells, podem veure com s'afecta a l'espectre del so original emès tots els efectes físics del propi cos del individu i amb aquesta diferència s'obté la HRTF que es pot utilitzar per filtrar qualsevol so i donar-li les mateixes característiques que les de l'individu amb qui s'ha realitzat l'aproximació



*Figura 15. Obtenció empírica de HRTF*

L'aproximació empírica té alguns desavantatges a l'hora d'obtenir les dades:

- És molt laboriós el treball a realitzar.
- Les condicions han de ser molt específiques i d'alta qualitat com poden ser les sales anecoiques o els aparells d'alta precisió com micròfons o altaveus que es necessiten. El que implica elevats costos.
- Hi ha punts on la captació és difícil, per elevacions molt baixes tals com per sota de  $-50^\circ$  que la infraestructura és complicada.

### **2.4.3.3- Bases de dades, matriu de respostes impulsional**

Quan es realitza una aproximació empírica es realitza una HRTF per a un azimuth i elevació concreta, o sigui pel cas quan una font sonora està situada en un punt concret de l'espai. Per tant s'han de tenir un parell de HRTF (una HRTF per l'oïda esquerra i una HRTF per l'oïda dreta) per a diversos punts de l'espai per tenir la resposta freqüencial general que tenen les oïdes d'una persona.

Les bases de dades són un conjunt d'arxius organitzats de forma lògica per elevació i azimuth. Normalment s'organitza en carpetes pels diferents nivells d'elevacions; i dintre de cada carpeta es troben les HRTF dels azimuths que cobreixen des de els 0° fins a envoltar a l'individu.

Hi ha dos tipus de manera de capturar les dades per realitzar la base de dades de les HRTF:

- Empíricament amb éssers humans.
- Empíricament a partir de maniquins.

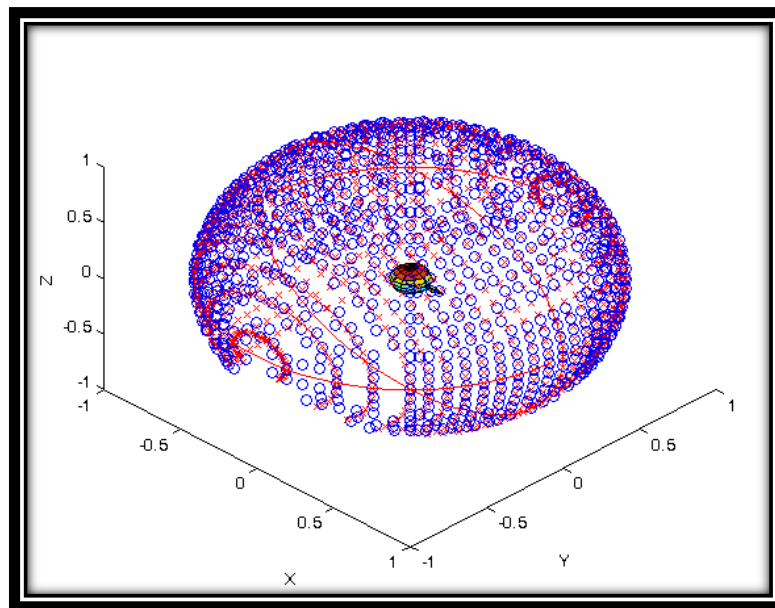
### **2.4.3.4- Mesures**

#### **Empíricament amb essers humans:**

Hi ha molt diverses maneres de realitzar les mesures empíriques per tal d'obtenir la resposta freqüencial d'un so tal qual arriba a la conca auditiva de les orelles d'un individu.

Per tal de dur a terme la base de dades de HRTF de tots els azimuths i elevacions els quals volem obtenir es fa un procediment físic en sales adequadament preparades acústicament, sales anecoiques, que consisteix en la col·locació d'uns petits micròfons d'alta precisió a l'entrada de la conca de les oïdes de l'individu els quals captaran els estímuls sonors situats a un punt concret de l'espai.

El procés consisteix a mostrejar una esfera al voltant del cap de l'individu. Els punts de mostreig tindran coordenades, azimuth "x" i elevació "y" en graus en radians.



**Figura 16. Punts de mesura per obtenir les HRTF**

En cada punt es crea un impuls sonor a través d'un altaveu amb un espectre freqüencial conegut, el més pla possible, que és captat per cada micròfon. L'espectre freqüencial captat és la HRIR, i fent la transformada de Fourier obtenim la HRTF a un azimuth "x" i elevació "y".



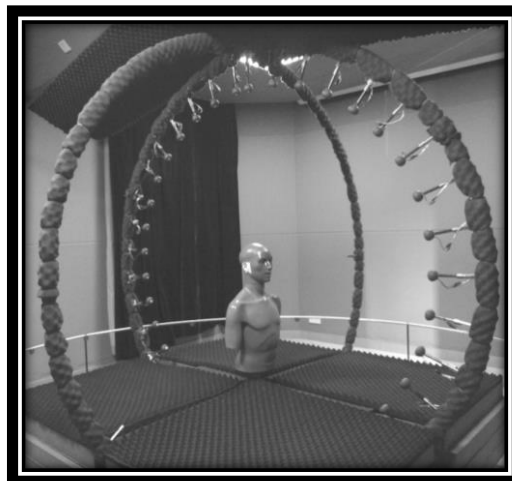
**Figura 17. Mostra el procés de captura amb un ésser humà**



### Empíricament a partir de maniquins:

El procés de captura amb un ninot és molt semblant, per no dir idèntic. Però en comptes d'utilitzar un cap humà, s'utilitza un ninot, que emula la fisiologia humana. El ninot potser bé torç o solament un cap.

El problema de realitzar empíricament les mostres amb un maniquí és que generalitzes una sèrie d'aspectes fisiològics humans amb un ninot i llavors no és la HRTF d'un humà, sinó semblant. Per tant no deixa de ser una aproximació, però una aproximació molt vàlida i útil per dur a terme projectes en el que has de generalitzar un software per a molt diverses persones i per tant no pots tenir una base de dades personalitzada per a tot tipus de persones, amb la càrrega de memòria que comportaria.



*Figura 18. Mostra el procés de captura amb ninot*

L'avantatge de fer-ho amb éssers humans és que pots recrear la captació de fisiologies concretes i molt diverses que tenim cada persona en particular, tal com podem trobar en la base de dades del IRCAM ([www.recherche.ircam.fr](http://www.recherche.ircam.fr)), diverses bases de dades capturades amb diferents individus i amb diferents característiques fisiològiques tals com l'amplada del cap, altura.... És una base de dades molt interessant però poc viable per utilitzar en un projecte, donat que sols amb la base de dades el projecte ja ocuparia molta memòria.

## 2.5- ffmpeg

"Ffmpeg" és una col·lecció de programes de lliure divulgació ("software" lliure) que conté una sèrie de llibreries que ens permeten manipular arxius d'àudio i vídeo. Ens permet descodificar, codificar, transcodificar, filtrar i crear nous arxius d'àudio i vídeo i funciona a través de línia de comandes. També pot convertir entre freqüències de mostreig arbitràries i canviar la mida de vídeo sobre la marxa. Ffmpeg pot llegir d'un nombre arbitrari d'arxius d'entrada (tals com arxius normals, fluxos de xarxa, dispositius d'enregistrament...), el flux d'entrada s'especifica amb la comanda "-i" (ffmpeg -i input.avi), i escriu en un nombre arbitrari d'arxius de sortida, el qual s'ha d'especificar mitjançant un nom d'arxiu de sortida sense format. Qualsevol cosa que es troba a la línia d'ordres que no pot ser interpretat com una opció es considera que és un nom de fitxer de sortida.

Hi ha una versió Matlab amb algunes funcions bàsiques, però la versió de línia de comandes de Windows també es pot utilitzar en Matlab mitjançant la funció "system".

System executa ordres del sistema operatiu i retorna la sortida, aquesta funció MATLAB crida al sistema operatiu per executar l'ordre especificada.

***status = system(command)***

Opcions bàsiques:

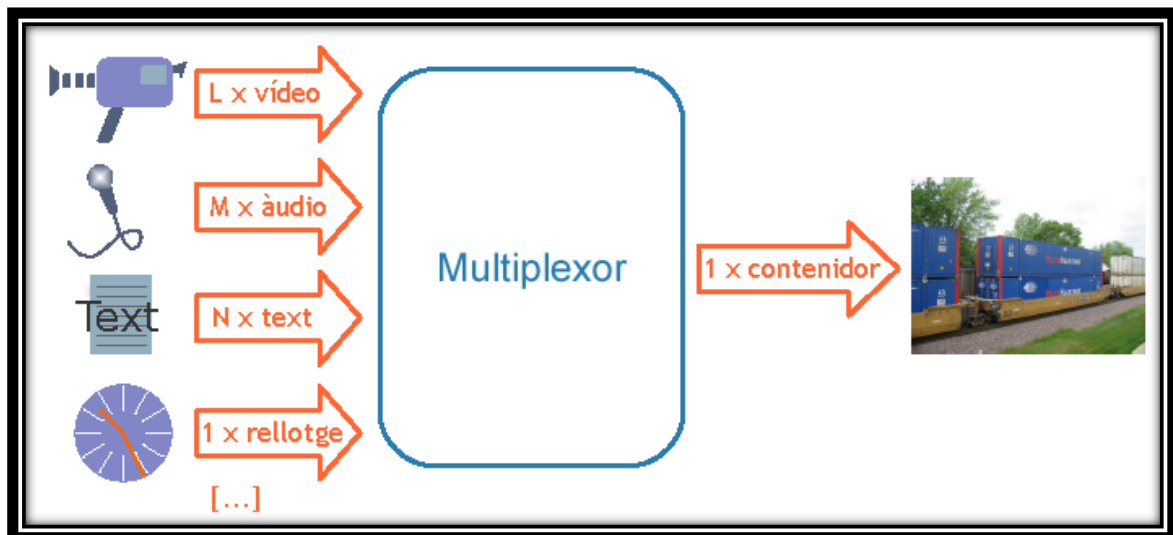
- **-i "ARXIU"**: Arxiu d'entrada
- **-map "ARXIU":pista**: Selecciona una pista concreta d'un dels arxius. **ARXIU** és el número assignat a un arxiu. Pots saber el numero de les pistes executant el programa sense especificar l'arxiu de sortida.
- **-n**: no sobre escriure cap arxiu. Si el nom d'arxiu de sortida és el mateix que el d'un arxiu que ja existeix per defecte pregunta si es vol que sigui sobreescrit.
- **-b:v \***: Tassa de bits (o bitrate) del vídeo (bits/s).
- **-codec:v "codec"**: Còdec de vídeo. Exemple: mpeg4
- **-r \***: Imatges per segon.
- **-s \*x\***: Dimensions del vídeo. Exemple: 320×240 (amplada per altura).

- **-aspect \*:\***: Aspecte. Exemple: 16:9. També pot ser un número decimal.
- **-vn**: Elimina la pista de vídeo.
- **-b:a \***: Tassa de bits (o bitrate) del àudio (bits/s).
- **-ar \***: Freqüència de l'àudio (Hz)
- **-codec:a "codec"** : Còdec d'àudio. Eexemple: mp3
- **-vol \***: Volum de l'àudio. 256=normal
- **-an**: Eliminar la pista d'àudio.
- **-sn**: Eliminar la pista de subtítols.
- **-threads \***: Indica el número de nuclis del processador que vols que s'utilitzin. Si el valor es 0, el programa seleccionarà el número de nuclis que vagin a proporcionar un millor resultat.

## 2.6- Contenedors multimèdia

Cada sistema digital, sigui un programa o un sistema operatiu, emmagatzema les dades de forma diferent dintre d'un arxiu d'àudio. Un arxiu d'àudio no sols conté la informació de totes les mostres de l'àudio digitalitzat, sinó que a més a més conté metadades a una capçalera que informa de diverses coses com per exemple com ha sigut generat el fitxer, nom, freqüència de mostreig, *bit rate*\*... Per tant un arxiu d'àudio està compost d'una capçalera més les dades de l'àudio pròpiament dit.

Un contenidor multimèdia és un tipus d'arxiu el qual té la funció d'emmagatzemar informació d'àudio, vídeo, imatges, metadades (com per exemple subtítols) i tot multiplexat d'una determinada manera. La utilització d'un tipus de contenidor multimèdia no implica en cap cas la utilització d'un còdec determinat.



*Figura 19. Esquema de creació d'un arxiu contenidor*

Existeixen diferents tipus de contenidors específics per a diferents tipus d'informació, mentre que hi ha d'altres compatibles per a la combinació de diferents tipus d'informació. Finalment per a la reproducció final d'aquests arxius el programa ha de saber simplement com desmultiplexar la informació que hi ha continguda dintre.

Lavors quan parlem amb la terminologia de "format" d'àudio o vídeo, estem parlant de la combinació dels següents conceptes:

- **Contenidor:** El contenidor és qui marca l'estructura de l'arxiu, on s'emmagatzemen les diverses estructures de dades, com s'intercalen. És extensible tant a un còdec d'àudio com a un de vídeo. S'utilitza per empaquetar el vídeo/àudio i els seus components (metadades) i s'identifica per una extensió d'arxiu, tals com .AVI, .mp4 o MOV.
- **Còdec:** Un còdec ("codificador / decodificador") és una forma de codificar àudio o vídeo en un corrent de bytes. És el mètode utilitzat per codificar el vídeo/àudio i és el principal determinant de la qualitat. Els còdecs són utilitzats dintre dels contenidors.

Podríem dir que en general els diferents formats de fitxers es diferencien en la manera que internament estructuren les dades. Cadascun té la seva pròpia manera de codificar la informació i té ratis de compressió diferents.

A continuació exposarem els formats que han intervingut directament en la realització d'aquest projecte.

## **2.7- Formats de vídeo**

Els formats de vídeo exposats són els formats de vídeo que accepta el software desenvolupat en aquest projecte. Han sigut escollits per què són uns dels formats més utilitzats en la distribució de vídeo multimèdia.

### **MKV(Matroska):**

Format d'estàndard obert per vídeo, àudio i subtítols. No és un còdec, és un contenidor que integra vídeo, àudio i subtítols dintre d'un mateix paquet permetent que no sols es pugui reproduir a un ordinador sinó a més dispositius amb suficient capacitat. Matroska no especifica un format de compressió, per tant, molt diversos còdecs poden ser utilitzats.

### **AVI:**

AVI (Audio Video Interleave) és un format contenidor d'àudio i vídeo llançat per Microsoft en 1992. Igual que el Matroska, es poden utilitzar diversos còdecs.

### **MP4:**

Format de contenidor multimèdia de llicència no oberta MPEG (Moving Picture Experts Group). Regulat per la ISO/IEC 14496-14:2003. A l'igual que els anteriors, es poden utilitzar diversos còdecs.

## 2.8- Formats d'àudio

Els formats d'àudio exposats són els formats que s'han utilitzat en aquest projecte. El format Ac3 multicanal ha sigut l'arxiu en el qual temporalment s'ha guardat totes les pistes d'àudio que s'extreuen d'un vídeo. Un cop tenim tot l'àudio contingut en un Ac3 llavors és quan s'utilitzen els arxius en format WAV per guardar, pista a pista de l'arxiu Ac3, cada una en un arxiu WAV diferent. Es va escollir WAV per què és un format contenidor d'àudio sense compressió i així minimitzar les pèrdues de qualitat en el procés.

### **WAV:**

Format d'àudio sense pèrdues, els arxius poden ser tant mono com estèreo a diverses resolucions i velocitats de mostreig, la seva extensió és .wav. Com que no té pèrdues de qualitat sol ser usat en l'àmbit professional.

### **AC3:**

Ac3 és el format utilitzat per la companyia Dolby Digital. Aquest tipus de format pot suportar Mono, estèreo i multicanal. Utilitza un format amb pèrdues, utilitza compressió perceptual, el que vol dir és que per aconseguir que l'arxiu pesi menys eliminen tot allò que no és audible per l'home, per tant elimines informació irrellevant i alleugereixes el pes de l'arxiu final.

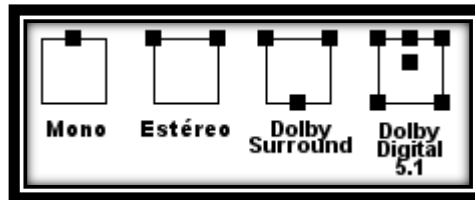
\*Bit rate es defineix com la quantitat de bits que es transmeten per unitat de temps (normalment en segons *bps*) a través d'un sistema de transmissió digital.

## 2.9- So envolvent 5.1

El so envolvent 5.1 o *Home cinema* és un sistema de so multicanal d'alta qualitat compost per 6 altaveus disposats al voltant de l'oient amb l'objectiu de fer-lo sentir que està al mig de l'acció d'una pel·lícula. La finalitat d'un sistema de so envolvent 5.1 és la d'apropar un sistema de so professional d'un cinema al saló d'una casa convencional.

Àudio multicanal fa referència a l'ús de múltiples pistes d'àudio per a reproduir un instant ambiental sonor. La terminologia de dues xifres separades per un punt decimal ens indica el

tipus de configuració d'altaveus per la qual es poden classificar (2.1, 5.1, 7.1, etc.). Depenent de la quantitat de pistes que s'utilitzin la terminologia varia, el primer dígit ens indica el nombre de pistes primàries que s'utilitzen en cada sistema (cada pista és reproduïda per un sol altaveu); Mentre que el segon dígit ens indica que la configuració porta incorporat un sistema de reproducció de baixes freqüències.

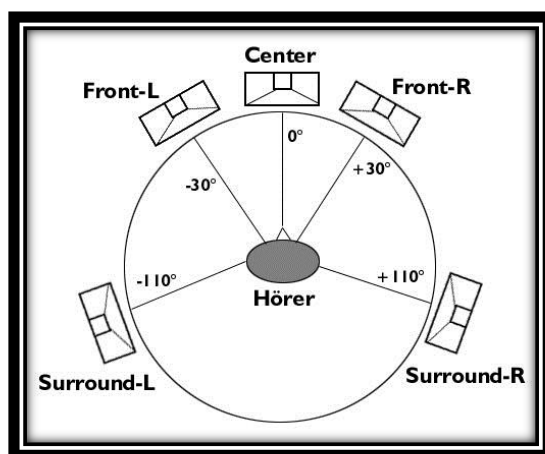


**Figura 20. Diferents tipus de configuració (Mono, estèreo, 2.1, 5.1)**

La distribució dels altaveus en un sistema 5.1 és molt important donat que afecta directament en la qualitat del sistema i en l'efecte d'immersió acústica i realisme pel qual destaquen aquest tipus de sistemes. Els altaveus frontals han d'estar a l'altura del cap, mentre que els trasers, *surround*, han d'estar un mica més elevat. El cas de l'altaveu que reproduïx la pista de greus pot estar situat a qualsevol punt de la sala, ja que la direccionalitat de les ones a baixes freqüències es comporten de forma omnidireccional.

Angularment els altaveus frontals dret i esquerre han d'estar, partint de què la cara de l'oient és l'angle 0°, entre 20° i 45°, el frontal central ha d'estar situa just davant de l'oient a 0° i els trasers entre 90° i 110°. Cap configuració és vàlida per a totes les persones, són unes pautes a seguir però els angles poden variar una mica, depenent de la fisiologia de cada persona pot ser més adient una configuració o una altra.

La forma en què estan distribuïts està ideada per què en conjunt facin l'efecte d'immersió auditiva en la pel·lícula. Dels 5 altaveus el frontal radia veu i diversos sons, els dos davanterers, esquerra i dret, radien bàsicament efectes i sons de tot tipus i els dos de darrere creen l'ambientació, s'encarreguen de donar la sensació que estàs al mig de l'acció. Després tenim l'altaveu de greus ".1" que emet tot els sons de baixa freqüència.



*Figura 21. Disposició d'un sistema 5.1 al voltant d'un oient*



## 3- Implementació

Amb la idea de programar petites funcions per al final compondre el programa final, el primer plantejament ha sigut el de fer un esquema general per veure les diferents fases per on passa el senyal digital i així desenvolupar petites funcions amb el fi d'unint totes aconseguir l'objectiu final, però que també fos senzill aïllar possibles problemes de funcionament.

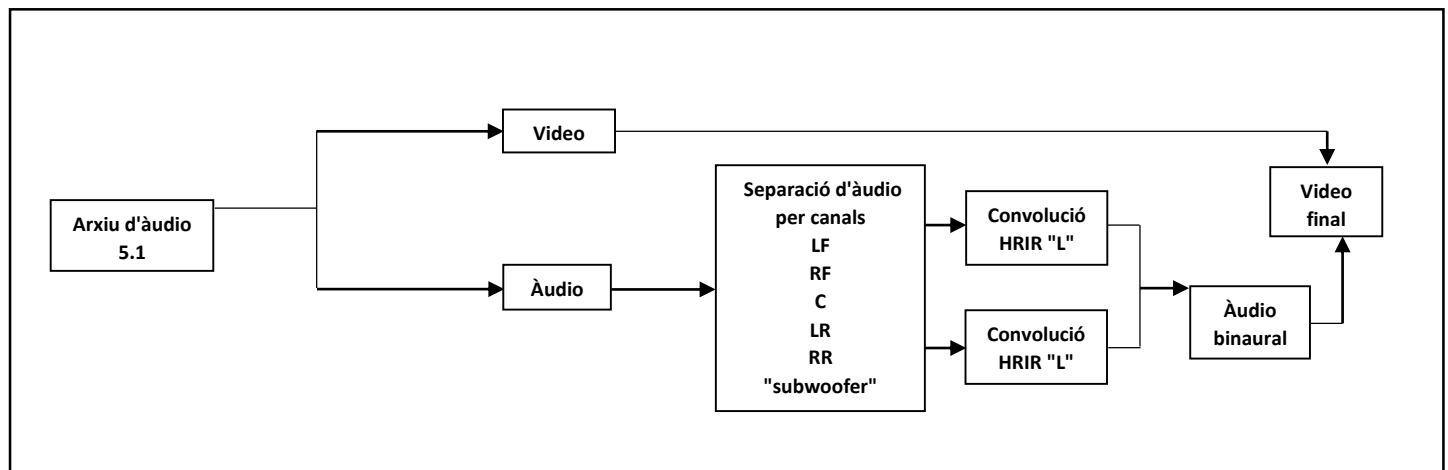
L'entorn de programació utilitzat ha sigut Matlab.

### 3.1- Matlab, per què?

MATLAB és un entorn molt potent per fer càlculs numèrics i a més és un llenguatge de programació molt semblant a d'altres llenguatges, però amb les seves particularitats. Creat per la companyia MathWorks, MATLAB permet moltíssimes coses com manipular matrius de forma molt senzilla, fer gràfics de les funcions i dades que s'estan manipulant, implementar algorismes, comunicar-se amb altres programes en altres llenguatges i crear interfícies d'usuari, utilitat molt pràctica i senzilla per fer programes a escala científica. El mateix Matlab conté una sèrie de funcions pre programades molt útils per la manipulació de dades, tals com les funcions convolució " $w = \text{conv}(u,v)$ " (funció utilitzada en el filtratge de l'àudio) o system " $status = \text{system}(command)$ " que ha sigut útil per fer servir la llibreria "ffmpeg".

Així que per aquesta facilitat per crear prototips de programes de forma ràpida i eficaç i per la quantitat d'avantatges en tractament de senyal hem escollit Matlab com entorn de programació i computació.

## 3.2- Esquema general del programa



*Figura 22. Funcionament esquemàtic del programa de conversió d'àudio multicanal a binaural*

## 3.3- 1ª Fase: extracció de l'àudio



*Figura 23. Taula de canals a extreure*

Per l'extracció de l'àudio han sigut claus dues funcions:

- **"ffmpegtranscode"**: Per extreure l'àudio del arxiu d'entrada "mkv", "avi" o "mp4" a un "ac3" multicanal.

- **"System"**(Matlab): Per poder executar comandes de programes aliens a Matlab. En aquest cas ens ha deixat utilitzar funcions de ffmpeg que no hi havia a la versió "ffmpeg" per Matlab com per exemple per extreure canal per canal tots els canals de l'arxiu "ac3".

Aquesta part del codi s'ha programat al script de la GUI ("GUI.m"), per la seva simplicitat i poques línies de codi:

```
ffmpegtranscode(fulpathname,'x.ac3', 'Range' , [0 d.duration] , 'Units' , 'seconds' , 'FastSearch' ,
'on' , 'VideoCodec' , 'none' );

system('ffmpeg -i x.ac3 -filter_complex channelsplit=channel_layout=5.1[FL][FR][FC][LFE][BL][BR] -map
"[FL]" front_left.wav -map "[FR]" front_right.wav -map "[FC]" front_center.wav -map "[LFE]" lfe.wav -
map "[BL]" back_left.wav -map "[BR]" back_right.wav');
```

**Figura 24. Codi extret de la funció GUI.m**

Després de l'execució d'aquestes línies de codi el que obteníem eren 6 arxius d'àudio corresponents a cada un dels canals que units formen la sensació de so envoltent 5.1

LF	Pista d'àudio frontal esquerra
RF	Pista d'àudio frontal dreta
C	Pista d'àudio central
LR	Pista d'àudio darrere esquerra
RR	Pista d'àudio darrere dreta
"Subwoofer"	Pista d'àudio de greus

**Figura 25. Índex dels canals d'un sistema 5.1**

Cal remarcar que el procés automàticament genera per cada canal d'àudio un arxiu en format "wav" per cada un dels canals (front\_left.wav, front\_right.wav, front\_center.wav, lfe.wav, back\_left.wav, back\_right.wav). És molt important remarcar que aquest arxius són "Mono", no "Stereo".

## 3.4- 2ªFase: Filtratge

És la part més important de tot el projecte donat que és l'apartat que li dona sentit a al projecte en sí mateix. En aquesta part del projecte es on s'han agut de programar més funcions per fer diverses tasques, és on filtrem l'àudio i obtenim l'àudio resultant de tots els filtratges.

Es van haver d'afrontar diversos reptes amb les HRTF. Primer es va haver de buscar una base de dades a internet suficientment fiables, veure com extreure la informació, veure com venia la HRTF (temps o freqüència) per finalment filtrar.

D'aquesta tasca s'encarrega la funció "filtr\_HRTF.m":

```
function x = filtr_HRTF(audio , FS , angle_source , head_width)
```

És la funció principal, el cor del programa. D'ella es deriven diverses funcions que s'aniran detallant i que estan adjuntes a l'annexa del projecte. Els parametres d'entrada són directa o indirectament necessaris pel càlcul de paràmetres. És important entendre que en aquesta funció entra un senyal "mono" i en surt un de binaural, amb dos canals un "L" i un "R".

- x : És el paràmetre que retorna la funció "filtr\_HRTF.m". És un senyal binaural.
- audio: És el senyal d'entrada a filtrar amb les HRIR.
- FS: És la freqüència de mostreig a la que està mostrejat el senyal "audio", paràmetre necessari per el càlcul del número de mostres que cal afegir a cada canal per tal d'aconseguir un retard entre oïdes (explicat amb més detall a l'annex 7.5) degut a la diferència d'arribada del so des de la font fins a cada oïda.
- angle\_source: És l'angle que hi ha entre la vertical (azimut 0°) i la font sonora.
- head\_width: Paràmetre que té incidència directa en el càlcul de la diferencia d'arribada del so des de la font fins a cada oïda.

La base de dades utilitzada pel desenvolupament del "software" va ser una base de dades creada per Bill Gardner(<http://alumni.media.mit.edu/~billg/>) i Keith Martin (<http://alumni.media.mit.edu/~kdm/>) de MIT media Lab (<http://www.media.mit.edu/>) que es l'institut de tecnologia de Massachusets.

<http://sound.media.mit.edu/resources/KEMAR.html>

Les característiques de la base de dades són les següents:

- Mesures realitzades amb KEMAR dummy-head microphone (maig 1994).
- Les mesures consisteixen en les respostes impulsional dels d'uns altaveus Realistic Optimus Pro 7 muntats a 1.4 metres del ninot KEMAR.
- Freqüència de mostreig 44.1 KHz.
- Un total de 710 posicions van ser mostrejades a elevacions des de -40 graus fins a +90 graus.



*Figura 26. Imatge d'un ninot KEMAR model KU 100 Neumann*

Com a la base de dades de que disposem tenim les respostes impulsional, o sigui que tenim la resposta en temps, teníem dos opcions:

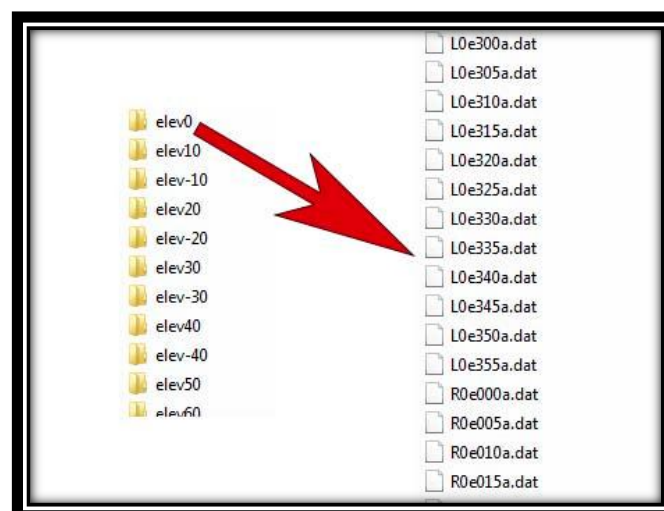
- Fer la Transformada de Fourier i passar de temps a freqüència i utilitzar la funció de Matlab per filtrar ("y=filter(a,b,x)").

- Filtrar el so original amb la resposta impulsional utilitzant la funció Matlab de convolució ("w=conv(u,v)").

La opció triada ha sigut la segona, filtrar el so fent la convolució de l'àudio amb la HRIR. Ha sigut la opció escollida degut a que és l'opció directa al ja estar la base de dades en funció del temps i no de la freqüència. En cas contrari s'hauria d'haver fet la transformada de Fourier de la HRIR per obtenir la HRTF i llavors fer el filtratge. Degut a això ens estalviem el càlcul computacional de la transformada de Fourier.

Un bon experiment d'optimització, que no s'ha pogut fer per manca de temps, seria el de veure si és més ràpid, o sigui si estalviem recursos computacionals filtrant amb la HRTF mitjançant la funció "filter" o utilitzant la HRIR mitjançant la funció "conv".

El primer pas en el procés de filtratge és el de carregar la base de dades adequada per les característiques que volem. Per poder entendre aquest procés és primordial entendre que la base de dades està classificada de forma lògica per elevació (de -40° a 90° en salts d'increment de 10° en 10°) i azimuth (de 0° a 355° en increment de 5° en 5°) i que de cada posició d'azimut hi ha una arxiu per la oïda esquerra i un altre per l'oïda dreta. L'elevació en aquest cas es fixa, s'ha suposat el cas ideal en què els altaveus han de quedar a l'altura del cap, elevació de 0°.



**Figura 27. Il·lustració de la base de dades**

Un cop la base de dades amb elevació  $0^\circ$  i azimuth adequat, es passa al procés de filtratge del senyal d'àudio. Aquest procés és simple de programar en Matlab gràcies a la toolbox que ens facilita molt els càlculs matemàtics i amb dues comandes (una filtrat per cada oïda). Conseqüentment de cada senyal que entra a la funció

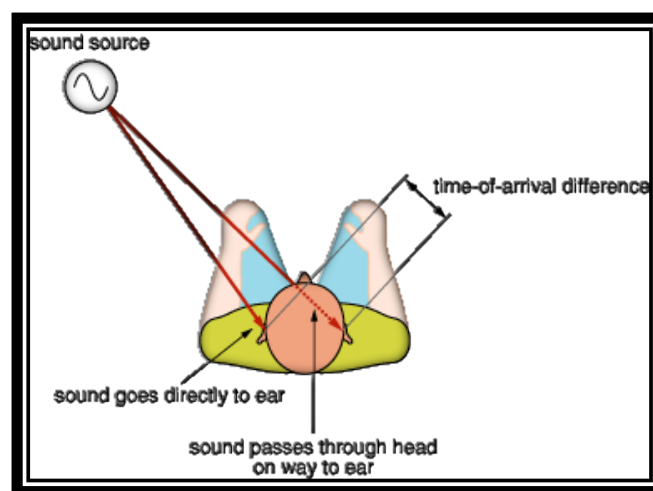
```
rightChannel=conv(audio,xr);  
leftChannel=conv(audio,xl);
```

**Figura 28.** *xr i xl son les HRIR del canal dret ("R") i del canal esquerre("L")*

Després d'aplicar la convolució tenim dos senyals que junts componen el senyal, del canal dret i del canal esquerre, binaural. Però en aquest punt els sons sonen al mateix temps tant en la oïda esquerre com en la dreta. Però a la vida real no passa això

### 3.5- Retard interaural

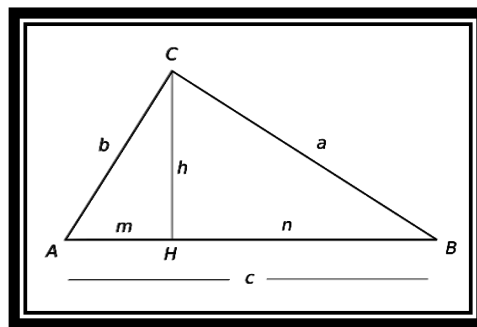
Com ja s'ha comentat anteriorment, el retard interaural (ITD-*interaural time delay*) és de màxima importància per a la detecció d'una font sonora a l'espai. Com ja hem dit quan el so arriba d'una font sonora amb un angle d'incidència, diferent d'azimut  $0^\circ$ , el so arriba avanç a l'oïda que rep el so directe que l'oïda que queda a l'ombra acústica creada pel cap.



**Figura 29.** *Esquema gràfic del retard temporal interaural*

Per tant, tenint coneixement d'aquest fenomen, s'ha agut de crear una funció, composta de diverses funcions per tal de calcular la diferència de temps interaural depenent de l'angle d'incidència de la font i amb aquesta informació aplicar un retard temporal a l'oïda que queda a l'ombra acústica del cap.

En resum, el so emès per una font a un punt de l'espai "C" triga un temps diferent en arribar a l'oïda "A" que el que triga a arribar a l'oïda "B" que esta situada en l'espai a un punt més llunyà de "C".



**Figura 30. Esquema geomètric pels càlculs interaurals**

O sigui que per dotar de realisme al nostre "software" cal aplicar un "delay" concret a un dels dos canals. Aquest canal serà el que, degut l'angle de la font respecte a l'oient, tingui una major distància entre la font i l'oïda del canal corresponent.

Per dur a terme aquesta tasca s'han creat tres funcions:

- DelaySamples.m: Aplica les mostres que cal concatenar al vector que conté l'àudio per aconseguir el retard adient depenent de l'ample del cap i de l'angle d'incidència de la font.
- calcul\_N\_samples: Calcula el nombre de mostres a aplicar depenent de la diferència entre la distància a la font del canal "L" i del canal "R".
- distancia\_LR: Calcula la distància que hi ha entre la font i cada una de les oïdes.

Com l'únic que ens interessa calcular és la diferència de temps que triga de més en arribar el so a una oïda que a l'altre el radi de distància entre l'oient i la font s'ha fixat a 1 m, fet que simplifica els càlculs.



El primer pas és calcular la distància que hi ha entre la font i cada una de les oïdes, aquest es un simple càlcul de geometria:

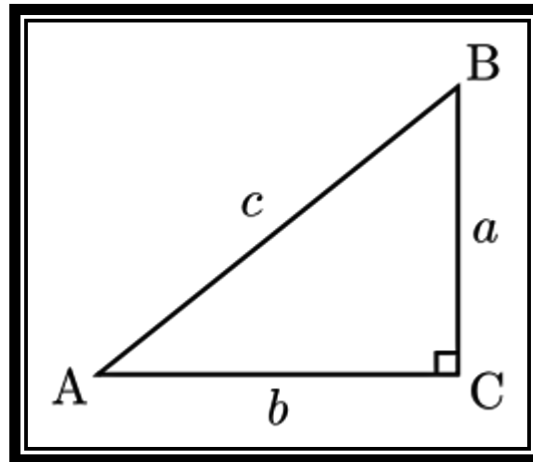


Figura 31. Esquema utilitzat en els càlculs

Oïda dreta: Si el punt "A" és el centre del cap, el punt "B" la font sonora i el "C" la oïda dreta ens queden distàncies de les quals dos són conegudes.

- a) És la distància entre la font i l'oïda i és desconeguda.
- b) És la distància entre la oïda i el centre del cap, per tant:  $b = Head_{with}/2$
- c) És la distància entre la font i el centre del cap que ho hem fixat a 1 metre.

$$a = c^2 + b^2 - 2 * b * c * \cos(angC)$$

Oïda esquerra: El càlcul és el semblant però canvien les incògnites

- a) És la distància entre la font i l'oïda i ja la tenim calculada.
- b) És la distància entre les dues oïdes:  $b = Head_{with}$
- c) És la distància entre la font i l'oïda esquerra que és desconeguda.

Així queda calculada la distància que hi ha entre la font i cada oïda, però al primer quadrant entre  $0^\circ$  i  $90^\circ$ . Per als demes és només trobar que el segon quadrant entre  $90^\circ$  i  $180^\circ$  les distàncies són iguals que al primer i els altres dos són inversos, la distància que surt a L entre  $0^\circ$  i  $180^\circ$  equival a la de R entre  $180^\circ$  i  $360^\circ$  i viceversa.

Després, com sabem que la velocitat a la que es desplaça el so és 343 m/s i la diferència de distància que hi ha entre l'espai que recorre el so fins arribar a l'oïda esquerra i el que triga a arribar a la dreta amb una resta, es pot calcular el temps de més que triga en arribar el so a l'oïda més llunyana de la font.

$$t = s/v$$

On "t" és el temps que triga de més, "s" és l'espai que recorre de més i "v" la velocitat del so.

Un cop tenim el temps amb la freqüència de mostreig ja podem calcular el número de mostres que cal aplicar amb valor 0 a l'inici del vector que volem que retardar "x" mostres en iniciar-se. És important concatenar el vector d'àudio que cal retardar amb el vector de 0 al inici, però a l'altre que no cal retardar se li ha de concatenar exactament el mateix vector de 0 però al final, per què tinguin la mateixa longitud i poder unir-los en un sol arxiu d'àudio de dos canals.

### 3.6- 3º Fase: Unir vídeo i àudio

Ara, un cop ja estan filtrats els 6 canal cal unificar-los en un de sol i definitiu arxiu binaural. Cal tenir en compte que la longitud de les mostres ha variat segons el canal, s'haurà de mirar quin té la longitud menor i sumar la mateixa quantitat de mostres, si no Matlab no permet fer sumes amb vectors de diferents longituds.

```
% Suma dels àudios.
m = [length(yrl), length(yfc), length(yrr), length(yfl), length(yfr), length(ylfe)];
m=min(m);
sum = yrl(1:m,:) + yfc(1:m,:) + yrr(1:m,:) + yfl(1:m,:) + yfr(1:m,:) + ylfe(1:m,:);
sum = normaliza_audio_stereo(sum);
```

**Figura 32. Codi extret de la funció GUI.m**

També es important normalitzar tot el vector d'àudio entre -1 i 1, ja que Matlab sempre treballa el so normalitzat i a l'hora de exportar l'àudio a un arxiu "WAV" estereo evitem el "*clipping*".

En aquest punt disposem ja d'un arxiu amb l'àudio de l'arxiu de vídeo multicanal per una banda i l'arxiu de vídeo en si mateix. Els passos seguits han sigut:

1. Preparar un arxiu de vídeo sense pista de àudio.
2. Unir en un arxiu l'àudio binaural i el vídeo.

Per dur a terme aquesta tasca s'ha tornat a utilitzar la biblioteca "ffmpeg", que també dona la possibilitat d'extreure el vídeo d'un arxiu amb una sola comanda, i després també ens permet unir un arxiu de vídeo i un d'àudio en un de sol amb les següents comandes.

```
system('ffmpeg -i orig.mkv -vcodec copy -an t.mkv ');
system('ffmpeg -i audioBIN.wav -i t.mkv videoOUT.mpg');
```

**Figura 33. Codi extret de la funció GUI.m**

## 3.6- GUI – Graphical User Interface

Matlab conté una funció per realitzar interfícies físiques de forma molt senzilla, utilitzant la comanda "*guide*" s'obre un diàleg per obrir una interfície ja realitzada o fer-ne una de nova. GUIDE es un entorn de programació visual per realitzar i executar programes que necessiten una entrada de dades. En definitiva és la manera en què l'usuari pot interactuar amb el programa.

La interfície física ha de constar d'uns bàsics per un correcte funcionament i a demés per a què l'usuari pugui introduir o seleccionar les opcions que vulgui provar.

### **5.1 File IN:**

A la part superior esquerra trobem el botó per seleccionar l'arxiu font d'entrada, que si el cliquem se'ns obra un diàleg per buscar la ruta on està l'arxiu que volem passar a binaural, l'objecte *static text* de l'esquerra del boto, mostrarà la ruta fins a l'arxiu seleccionat.

### **5.1 File OUT:**

Just a sota tenim el botó per indicar la ruta on volem emmagatzemar l'arxiu resultant binaural, en cas de no seleccionar cap ruta de sortida la ruta per defecte serà l'arrel del programa. L'objecte *static text* de l'esquerra del boto, mostrarà la ruta d'emmagatzemament.

### **Panell:**

Al panell podem seleccionar 3 coses que afecten directament al resultat del vídeo final. Primer l'amplada del cap, el qual afecta directament en el càlcul del retard interaural, després podem indicar l'angle d'incidència de la font tant dels altaveus frontals com dels altaveus trasers.

### **Gràfic:**

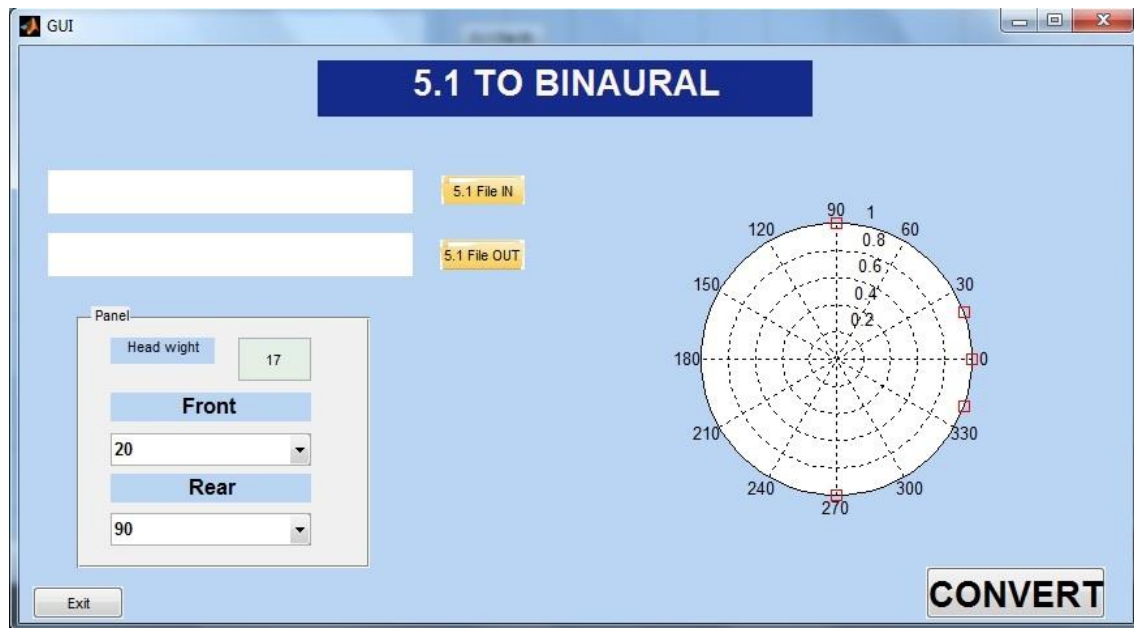
El gràfic polar ajuda visualment a veure com seria la disposició final de uns altaveus amb els angles escollits.

### **Convert:**

És el botó d'execució del programa.

### **Exit:**

És el botó per sortir del programa.



*Figura 34. Mostra de la interfície física del programa*

## **4 – Estudi d'optimització de la posició**

### **virtual dels altaveus**

La fisiologia humana és molt diversa i variada. Com ja s'ha comentat amb anterioritat la fisiologia de cada individu intervé directament en la manera en què cadascú escolta i pot localitzar el so a l'espai. La forma concreta del pavelló auditiu (pinna) que en cadascú és diferent, les reflexions del so en el tors de cada individu i la forma del cap, entre d'altres, modifiquen l'espectre freqüencial que rebem a les nostres oïdes i que ha sigut emès per una font en un punt de l'espai.

Per exemple com hem pogut veure a l'apartat de localització del so a l'espai un dels aspectes més importants són el retard temporal interaural que està directament lligat a la fisiologia del cap i a l'angle d'incidència de la font i la diferència interaural de nivell, també lligada a la fisiologia del cap el qual crea una ombra acústica a l'oïda més llunyana.

El que volem trobar fent aquest estudi és l'angle amb menor variabilitat entre diversos individus, el que vol dir trobar l'angle amb menor dispersió i implica menor desviació estàndard. Si trobéssim l'angle de menor variabilitat, tant en l'angle d'incidència dels altaveus tarsers com frontals, trobaríem l'angle més estàndard per a l'usuari. En altres paraules trobaríem quina és la millor opció a escollir com a angle d'incidència frontal i traser per a la majoria de persones.

Per dur a terme aquest estudi hem utilitzat les HRTF de diverses persones, amb fisiologies corporals diferents, per tal d'intentar trobar quin dels angles d'incidència és el més estàndard, el que té menys índex de dispersió. Hem comparat les HRTF de diferents persones en elevació 0° i azimuths 15°, 30°, 45° (en el cas dels angles d'incidència dels altaveus frontals) i 120°, 135°, 150° (en el cas dels angles d'incidència dels altaveus trasers).

El procés per dur a terme l'estudi és el de descarregar una base de dades de HRIR de diverses persones, extreure de cada individu la informació dels angles d'incidència que volem i després aplicar-les la transformada de Fourier per tal d'obtenir la HRTF i poder fer la desviació estàndard entre tots els individus a totes les freqüències, per elevació  $0^\circ$  i azimuts  $15^\circ$ ,  $30^\circ$ ,  $45^\circ$  i  $120^\circ$ ,  $135^\circ$ ,  $150^\circ$ . Amb els resultats s'ha fet una gràfica de la desviació estàndard en funció de la freqüència per visualitzar els resultats. Els càlculs matemàtics s'han realitzat en Matlab (annex 6.6) i el tractament de les dades per la realització de les gràfiques i poder visualitzar els resultats s'ha realitzat amb Excel.

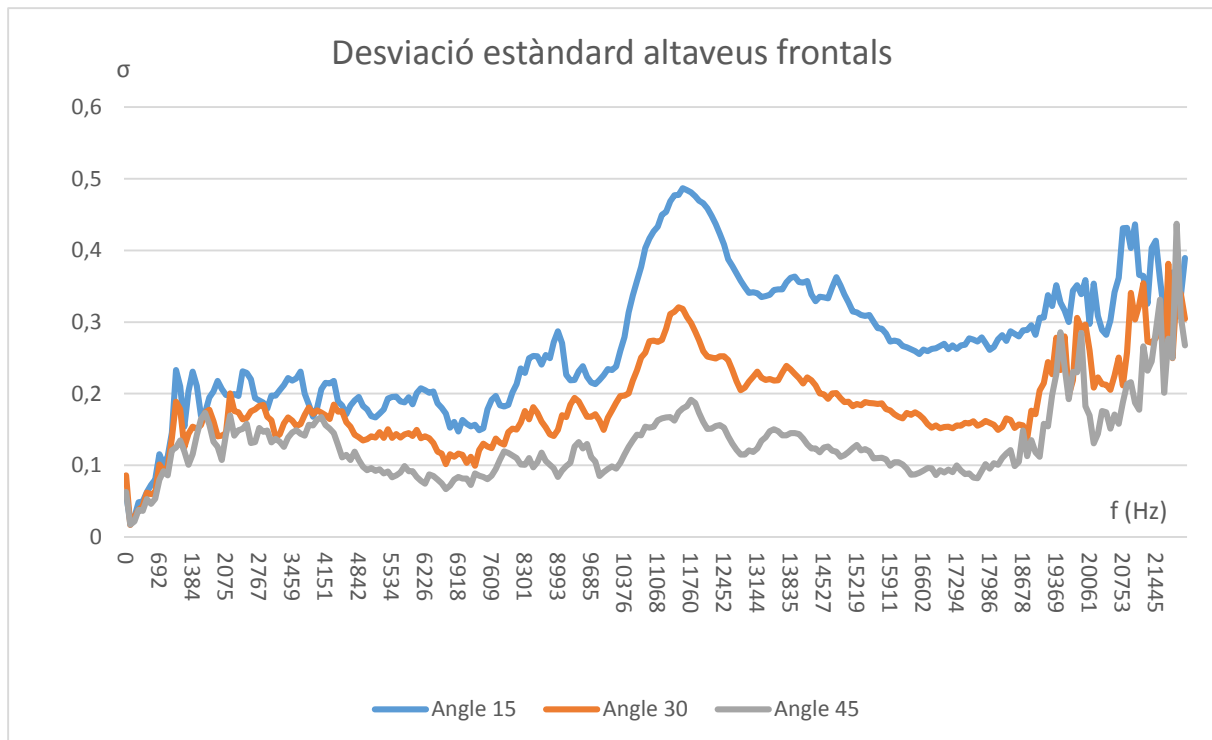
La desviació estàndard és una mida de dispersió la qual ens indica com és de diferent la mostra respecte de la mitja aritmètica. Així sabem com una mostra poden tendir a allunyar-se els valors respecte de la mitjana.

La base de dades descarregada ha sigut creada per l'IRCAM centre Pompidou (Institut de recerca i coordinació acústic/musical) a París, França (<http://www.ircam.fr/>) i la companyia AKG acústics (<http://www.akg.com/>). A la web de descàrrega, es pot trobar un total de 51 HRIR de 51 persones morfològicament diverses. La informació de cada individu ve codificada en arxius d'àudio ".wav" o en arxius Matlab ".mat", cada arxiu Matlab és un *struct* on queda classificada la informació. Els azimuts i les elevacions estan classificats cada  $15^\circ$  ( $0^\circ, 15^\circ, 30^\circ \dots$ ).

## **4.1- Altaveus frontals**

Com podem observar a la gràfica de la figura 34, les gràfiques de la desviació estàndard a  $15^\circ$  i  $30^\circ$  és clarament major que a  $45^\circ$ , per tant es desvien molt més de la mitjana aritmètica. Això vol dir que les funcions de transferències són més semblants entre tots els individus quan la font està situada a elevació  $0^\circ$  i azimuth  $45^\circ$ . Si analitzem la gràfica de la desviació estàndard, a elevació  $0^\circ$  i azimuth  $45^\circ$ , podem observar com entre les freqüències baixes i 18 KHz la funció és molt estable, això vol dir que per a baixes freqüències les HRTF no es dispersen tant com a partir de 18 KHz fins a 22 KHz, zona on

la inestabilitat és major i existeix major diferència en la resposta freqüencial dels individus entre si.



**Figura 35. Gràfica desviació estàndard en funció de la freqüència dels altaveus frontals**

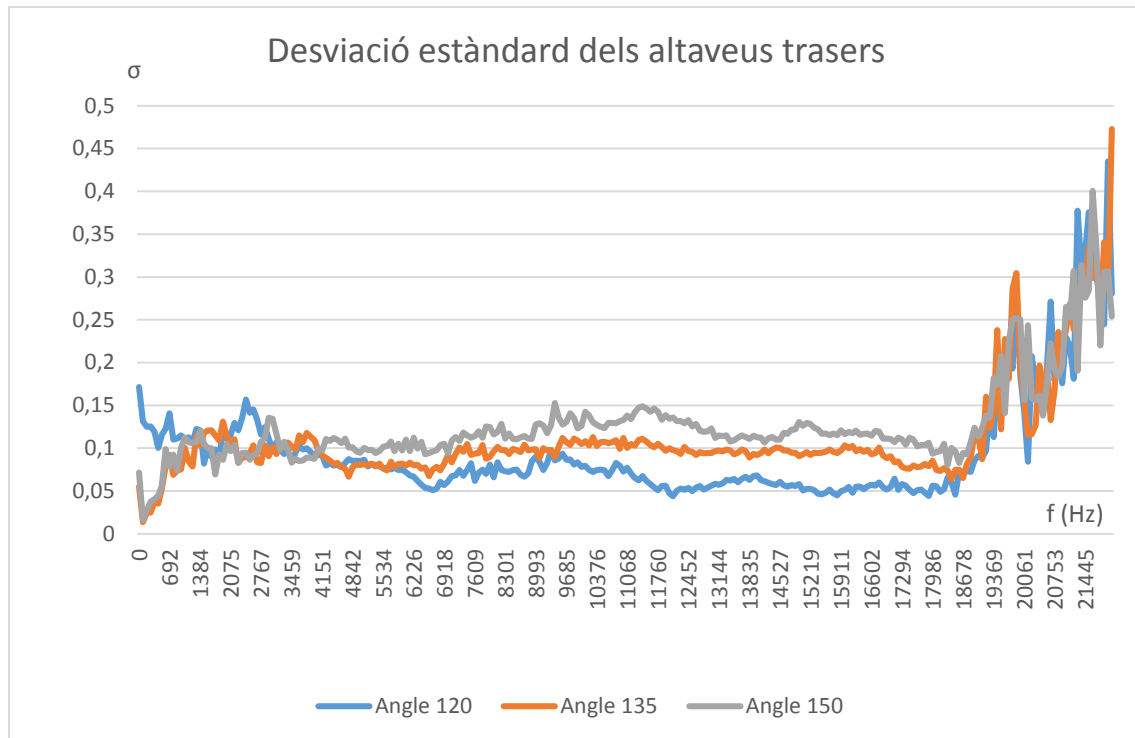
Els resultats obtinguts ens mostren que si s'escull 45° com a opció per l'angle d'incidència dels altaveus frontals el resultat majoritàriament ha de ser més òptim donat que la desviació estàndard per cada freqüència és menor en general. Per tant vora els 45° un número més elevat de gent obtindrà una millor escolta binaural.

## **4.2- Altaveus trasers**

En el cas dels angles de les fonts traseres, (figura 35) podem observar com no hi ha grans diferències entre les posicions d'azimut 120°, 135° i 150°. Com no hi ha una dispersió tan acusada com en el cas dels altaveus frontals podríem dir que en aquest cas cap de les



tres opcions és clarament més comuna en tots els individus. Es pot apreciar també que a partir de 18 KHZ les HRTF es dispersen molt de la mitjana aritmètica.



**Figura 36. Gràfica desviació estàndard en funció de la freqüència dels altaveus trasers**

Els resultats obtinguts ens mostren que l'angle d'incidència dels altaveus trasers és pràcticament indiferent. El que vol dir és que l'usuari pot escollir lliurement l'angle d'incidència que vulgui, que l'escolta binaural no s'hauria de veure afectada de forma notable.

### **4.3- Anàlisi subjectiu dels resultats**

En aquest apartat el que es pretén és fer un assaig subjectiu per tal de verificar, subjectivament, si els resultats anteriors realment apreciats a l'escolta real amb àudios creats amb codi extret del programa base del nostre projecte.

Per dur a terme aquest assaig crearem 9 arxius d'àudio amb totes les possibilitats d'angles amb què hem realitzat l'estudi teòric d'optimització de la posició virtual dels

altaveus. Per tal de crear-los, s'ha escrit un script Matlab, "scrip\_analisis.m" adjuntat a l'apartat d'annexes, el qual ens crea un arxiu "WAV" de 30 segons binaural extret d'un vídeo amb so Dolby multicanal.

S'ha dividit l'escolta en 3 grups l'anàlisi en funció dels angles d'incidència:

- Angle frontal 15°
  - Angle traser 120°
  - Angle traser 135°
  - Angle traser 150°
- Angle frontal 30°
  - Angle traser 120°
  - Angle traser 135°
  - Angle traser 150°
- Angle frontal 45°
  - Angle traser 120°
  - Angle traser 135°
  - Angle traser 150°

De cada grup s'ha de veure quin és el que fa sentir a l'oient més en el mig de l'acció i escollir-ne un. Un cop tinguem un escollit de cada grup, s'ha d'escollir dels tres millors quin és el millor i comparar si els resultats de l'anàlisi de dispersió de l'apartat anterior és realment correcte i si les diferències són realment perceptibles.

A la figura 36 es troben els meus resultats, els quals no coincideixen amb els resultats anteriors, però subjectivament la diferència entre l'angle d'incidència frontal a 30° o 45° són pràcticament inapreciables.

Azimut frontals	Azimut trasers	Grup	General
15	120		
	135	x	
	150		
30	120		X
	135		
	150	x	
45	120		
	135	x	
	150		

*figura 37. Taula de resultats de l'anàlisi subjectiu*

Els resultats de l'apartat anterior mostren el millor angle a escollir. Aquests resultats estan basats en el que és millor per a un nombre gran de persones, però no el resultat millor per a cada persona. Per tal de què l'estudi subjectiu fos significatiu i es pogués comparar amb els resultats de l'apartat anterior, s'hauria d'haver realitzat l'escolta subjectiva en un nombre gran de persones. Estudi interessant per a futurs projectes.

## **5 – Conclusions**

S'ha realitzat un programa que extreu l'àudio multicanal d'un vídeo, separa tots els canals en individuals diferents Mono, els filtra individualment mitjançant les HRTF, retorna l'àudio binaural (dos únics canals) juntament amb el vídeo inicial en un únic arxiu.

Això permet poder gaudir de l'àudio binaural (sensació d'estar escoltant 5.1) amb uns auriculars convencionals sense la necessitat de tenir un sistema de 6 altaveus ni un descodificador per crear la il·lusió d'estar al mig de la escena de la pel·lícula auditivament parlant. Aquest software permet portar a un dispositiu mòbil un sistema *surround* .

També s'ha buscat quin és l'angle d'incidència més òptim per la col·locació d'un sistema d'altaveus 5.1 mitjançant una base de dades de HRTF de molt diverses persones.

Un cop finalitzat el projecte, podríem dir que això és només la punta de l'*iceberg*, és una temàtica prou interessant per a desenvolupar molt diversos projectes en àmbits molt dispars però interessants al mateix temps. Encara que computacionalment encara és complicat, és un àmbit molt interessant.

Encara que no s'ha aconseguit un software molt eficient (el càlcul computacional és gran i es triga molt a aconseguir fer el procés d'extracció), filtratge i unió l'objectiu bàsic ha sigut assolit. S'ha fet un programa que hi entra un arxiu amb 6 canals d'àudio i en surt un que és simplement estèreo però mantenint l'espacialitat que s'aconsegueix amb el multicanal.

Futures extensions podrien ser l'aplicació del so binaural, no sols en el cinema, sinó també en altres aspectes com podrien ser els videojocs, l'ajuda a discapacitats invidents

o la realitat virtual. Partint de la base d'aquest projecte només seria canviar l'aplicació cap a la qual es vol enfocar.

## 6- Bibliografia

1. Equipos de sonido McGraw Hill
2. [http://rabfis15.uco.es/lvct/tutorial/1/paginas%20proyecto%20def/\(4\)%20efectos%20del%20ruido/fisiologia%20de%20la%20audicion.htm](http://rabfis15.uco.es/lvct/tutorial/1/paginas%20proyecto%20def/(4)%20efectos%20del%20ruido/fisiologia%20de%20la%20audicion.htm)
3. <http://www.scielo.org.co/pdf/rlps/v41n2/v41n2a05.pdf>
4. <https://www.ffmpeg.org/ffmpeg.html>
5. <http://es.mathworks.com>
6. Apunts de processament digital d'àudio I. Esquerre
7. <http://www.dolby.com/in/en/technologies/dolby-digital.html>
8. <https://library.rice.edu/services/dmc/guides/video/VideoFormatsGuide.pdf>
9. <http://www.matroska.org/>
10. COREY I. CHENG AND GREGORY H. WAKEFIELD, Introduction to HRTF, J Audio Eng Soc, Vol 49, No. 4, 2001 April
11. Ralph Algazi and Richard O. Duda, Approximating the head-related transfer function using simple geometric models of the head and torso, J. Acoust. Soc. Am. 112 (5), Pt. 1, Nov. 2002
12. Fundamentals of Acoustics, apunts màster d'acústica etseiat.

## 7- Anexes

En aquest apartat quedarà exposat, el codi integra de les funcions creades per dur a terme aquest projectes.

### 7.1- "gui.m"

Aquesta és la funció en la que queda programada la interfície física.

```
function varargout = GUI(varargin)
% GUI MATLAB code for GUI.fig
%
%   GUI, by itself, creates a new GUI or raises the existing
%   singleton*.
%
%   H = GUI returns the handle to a new GUI or the handle to
%   the existing singleton*.
%
%   GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in GUI.M with the given input
%   arguments.
%
%   GUI('Property','Value',...) creates a new GUI or raises the
%   existing singleton*. Starting from the left, property value
%   pairs are
%   applied to the GUI before GUI_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property
%   application
%   stop. All inputs are passed to GUI_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
%   only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI
```

```
% Last Modified by GUIDE v2.5 01-Apr-2015 19:24:49

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @GUI_OpeningFcn, ...
                  'gui_OutputFcn',  @GUI_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUI is made visible.
function GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to GUI (see VARARGIN)


% Choose default command line output for GUI
handles.output = hObject;


% Update handles structure


    %           Aquí posem les icones i visualitzem el plot de les
opcions dels
%           angles que hi ha per defecte:
```



```

%               Es carreguen les imatges i es superposen sobre els
botons "ExaminaBut" i "Savepath_but"

guidata(hObject, handles);
[a,map]=imread('folder.jpg');
[r,c,d]=size(a);
x=ceil(r/35);
y=ceil(c/85);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;
set(handles.ExaminaBut,'CData',g);

guidata(hObject, handles);
[a,map]=imread('folder.jpg');
[r,c,d]=size(a);
x=ceil(r/35);
y=ceil(c/85);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;
set(handles.savepath_but,'CData',g);

%               Aquí obtenim els valors del menu desplegable i fem
el
%               polar plot:
selected_cmd = get(handles.PopRear,'Value');
switch selected_cmd
    case 1
        angle_source_rear_right = 90;
    case 2
        angle_source_rear_right = 95;
    case 3
        angle_source_rear_right = 100;
    case 4
        angle_source_rear_right = 105;
    otherwise
        angle_source_rear_right = 110;
end

angle_source_rear_right_rad = deg2rad(angle_source_rear_right);

```

```

    angle_source_rear_left_rad = degtorad(360-
angle_source_rear_right);

    selected_cmd = get(handles.PopFront, 'Value');
    switch selected_cmd
        case 1
            angle_source_front_right = 20;
        case 2
            angle_source_front_right = 25;
        otherwise
            angle_source_front_right = 30;
    end

    angle_source_front_right_rad =
degtorad(angle_source_front_right);%Paso de graus a radians per el
polar plot
    angle_source_front_left_rad = degtorad(360-
angle_source_front_right);

    rho = [1 1 1 1 1];
    theta = [angle_source_rear_right_rad angle_source_rear_left_rad
angle_source_front_right_rad angle_source_front_left_rad degtorad(0)];
    polar(theta,rho, 'rsquare')
    axis('square')

% UIWAIT makes GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = GUI_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

```

```

varargout{1} = handles.output;

% --- Executes on selection change in PopFront.
function PopFront_Callback(hObject, eventdata, handles)
% hObject      handle to PopFront (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns PopFront
contents as cell array
%          contents{get(hObject,'Value')} returns selected item from
PopFront

% Aquest és el menu desplegable per escollir l'angle d'incidència dels
altaveus frontals
    selected_cmd = get(handles.PopRear,'Value');%Rescatas sel valor
seleccionat
    switch selected_cmd
        case 1
            angle_source_rear_right = 90;
        case 2
            angle_source_rear_right = 95;
        case 3
            angle_source_rear_right = 100;
        case 4
            angle_source_rear_right = 105;
        otherwise
            angle_source_rear_right = 110;
    end

    angle_source_rear_right_rad = degtorad(angle_source_rear_right);
%Paso de graus a radians per el polar plot
    angle_source_rear_left_rad = degtorad(360-
angle_source_rear_right);

```

```

    selected_cmd = get(handles.PopFront, 'Value'); %Rescatas sel valor
seleccionat
    switch selected_cmd
        case 1
            angle_source_front_right = 20;
        case 2
            angle_source_front_right = 25;
        otherwise
            angle_source_front_right = 30;
    end

    angle_source_front_right_rad =
degtorad(angle_source_front_right); %Paso de graus a radians per el
polar plot
    angle_source_front_left_rad = degtorad(360-
angle_source_front_right);

    rho = [1 1 1 1 1];
    theta = [angle_source_rear_right_rad angle_source_rear_left_rad
angle_source_front_right_rad angle_source_front_left_rad degtorad(0)];
    polar(theta, rho, 's')

% --- Executes during object creation, after setting all properties.
function PopFront_CreateFcn(hObject, eventdata, handles)
    % hObject    handle to PopFront (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    empty - handles not created until after all
CreateFcns called

    % Hint: popupmenu controls usually have a white background on
Windows.

    %         See ISPC and COMPUTER.
    if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
        set(hObject, 'BackgroundColor', 'white');
    end

```

```
% --- Executes on selection change in PopRear.
function PopRear_Callback(hObject, eventdata, handles)
% hObject    handle to PopRear (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns PopRear
%         contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
%         PopRear

% Aquest és el menu desplegable per escollir l'angle d'incidència dels
% altaveus trasers:
    selected_cmd = get(handles.PopRear, 'Value'); %Rescatas sel valor
seleccionat
    switch selected_cmd
        case 1
            angle_source_rear_right = 90;
        case 2
            angle_source_rear_right = 95;
        case 3
            angle_source_rear_right = 100;
        case 4
            angle_source_rear_right = 105;
        otherwise
            angle_source_rear_right = 110;
    end

    angle_source_rear_right_rad =
degtorad(angle_source_rear_right); %Paso de graus a radians per el
polar plot
    angle_source_rear_left_rad = degtorad(360-
angle_source_rear_right);

    selected_cmd = get(handles.PopFront, 'Value');
    switch selected_cmd
        case 1
            angle_source_front_right = 20;
        case 2
```

```

        angle_source_front_right = 25;
    otherwise
        angle_source_front_right = 30;
    end

    angle_source_front_right_rad =
deg2rad(angle_source_front_right);%Paso de graus a radians per el
polar plot
    angle_source_front_left_rad = deg2rad(360-
angle_source_front_right);

    rho = [1 1 1 1 1];
    theta = [angle_source_rear_right_rad angle_source_rear_left_rad
angle_source_front_right_rad angle_source_front_left_rad deg2rad(0)];
    polar(theta,rho,'s')

% --- Executes during object creation, after setting all properties.
function PopRear_CreateFcn(hObject, eventdata, handles)
    % hObject    handle to PopRear (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    empty - handles not created until after all
CreateFcns called

    % Hint: popupmenu controls usually have a white background on
Windows.
    %         See ISPC and COMPUTER.
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

% --- Executes on button press in ExaminaBut.
function ExaminaBut_Callback(hObject, eventdata, handles)
    % hObject    handle to ExaminaBut (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)

```

```
%      Amb aquest botó es controla la cerca per guardar la direcció de
l'arxiu d'entrada:
    [filename , pathname] =
uigetfile({'*.avi'; '*.mp4'; '*.mkv'; '*.*'}, 'Select the VIDEO file');
    set(findobj('Tag', 'PathText'), 'String', [pathname filename]);

% --- Executes when figure1 is resized.
function figure1_ResizeFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in togglebutton1.
function togglebutton1_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of togglebutton1

% --- Executes on button press in ExitBut.
function ExitBut_Callback(hObject, eventdata, handles)
% hObject    handle to ExitBut (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close(gcf)

function PathText_Callback(hObject, eventdata, handles)
% hObject    handle to PathText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```

% Hints: get(hObject,'String') returns contents of PathText as text
%         str2double(get(hObject,'String')) returns contents of
PathText as a double
% PathTexB = findobj(gcbf,'Tag','PathText');

% set(handles.PathText,'String',pathname)

% --- Executes during object creation, after setting all properties.
function PathText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PathText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.

% set(handles.PathText,'String',pathname)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function HW_IN_Callback(hObject, eventdata, handles)
% hObject    handle to HW_IN (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of HW_IN as text
%         str2double(get(hObject,'String')) returns contents of HW_IN
as a double

% --- Executes on button press in savepath_but.
function savepath_but_Callback(hObject, eventdata, handles)
% hObject    handle to savepath_but (see GCBO)

```



```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
folder_name = uigetdir;
set(findobj('Tag','savepath_text'),'String',[folder_name]);

% --- Executes on button press in DoBut.
function DoBut_Callback(hObject, eventdata, handles)
% hObject handle to DoBut (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Aquest és el botó on succeïx tot, és on es desencadena tot el procés
del programa:
HW = get(handles.HW_IN,'String');%Obtenim l'amplada del cap
fulpathname = get(handles.PathText,'String');%Obtenim la direcció
del arxiu d'entrada
selected_cmd = get(handles.PopRear,'Value');%Obtenim els Angles
d'incidència
switch selected_cmd
    case 1
        angle_source_rear_right = 90;
    case 2
        angle_source_rear_right = 95;
    case 3
        angle_source_rear_right = 100;
    case 4
        angle_source_rear_right = 105;
    otherwise
        angle_source_rear_right = 110;
end
angle_source_rear_left = 360-angle_source_rear_right;

selected_cmd = get(handles.PopFront,'Value');
switch selected_cmd
    case 1
        angle_source_front_right = 20;
    case 2
```

```

        angle_source_front_right = 25;
    otherwise
        angle_source_front_right = 30;
    end
    angle_source_front_left = 360-angle_source_front_right;

%     ffmpegsetup
% Un cop obtingudes les dades necessàries esborrem els arxius que hi
puguin quedar d'un procés
% que s'hagi quedat a mitges anteriorment:
    delete('X.ac3');

delete('front_right.wav','front_left.wav','front_center.wav','back_left.wav',
'back_right.wav','lfe.wav');

% Aquí primer extraiem el àudio del vídeo en un únic arxiu en format
AC3, el qual després
% li extraurem un per un cada canal.
    d=ffmpeginfo(fulpathname);
    ffmpegtranscode(fulpathname,'x.ac3', 'Range' , [0 d.duration] ,
'Units' , 'seconds' , 'FastSearch' , 'on' , 'VideoCodec' , 'none' );
    system('ffmpeg -i x.ac3 -filter_complex
channelsplit=channel_layout=5.1[FL][FR][FC][LFE][BL][BR] -map "[FL]"
front_left.wav -map "[FR]" front_right.wav -map "[FC]"
front_center.wav -map "[LFE]" lfe.wav -map "[BL]" back_left.wav -map
"[BR]" back_right.wav');
    msn='audio extraction done'

%Llegim els 5 canals principals Front, 2xFronts & 2xRear
[x , FS] = audioread('back_left.wav');%Llegir l'àudio
yrl = filtr_HRTF(x , FS , angle_source_rear_left , HW);%Filtrem
l''audio
yrl = normaliza_audio_stereo(yrl);%Normalitzem l'àudio entre -1 i
1

[x , FS] = audioread('back_right.wav');
yrr = filtr_HRTF(x , FS , angle_source_rear_right , HW);
yrr = normaliza_audio_stereo(yrr);

```

```

d='rear done'

[x , FS] = audioread('front_center.wav');
yfc = filtr_HRTF(x , FS , 0 , HW);
yfc = normaliza_audio_stereo(yfc);

[x , FS] = audioread('front_left.wav');
yfl = filtr_HRTF(x , FS , angle_source_front_left , HW);
yfl = normaliza_audio_stereo(yfl);

[x , FS] = audioread('front_right.wav');
yfr = filtr_HRTF(x , FS , angle_source_front_right , HW);
yfr = normaliza_audio_stereo(yfr);
dd='front done'

[x , FS] = audioread('lfe.wav');
dd = 'se empieza el script de graves'

% "greus" es una un pequeño script que crea una unica fuente de
% graves binaural como si viniesen los graves de los 5 puntos
% diferentes.
ylfe = greus(x , FS , HW , angle_source_front_left ,
angle_source_front_right , angle_source_rear_left ,
angle_source_rear_right);
dd = 'acaba el script de graves'

% Suma dels àudios.
m = [length(yrl) , length(yfc) , length(yrr) , length(yfl) ,
length(yfr) , length(ylfe)];
m=min(m);
sum = yrl(1:m , :) + yfc(1:m , :) + yrr(1:m , :) + yfl(1:m , :) +
yfr(1:m , :) + ylfe(1:m , :);

sum = normaliza_audio_stereo(sum);%Normalització del àudio
resultant
dest_path = get(handles.savepath_text,'String');%Obtenim on s'ha
de guardar

```

```
% Creem l'arxiu WAV final que unirem al vídeo.
audiowrite([pwd '\audioBIN.wav'],sum,FS);

% Eliminem els arxius residu creats en el procés
delete('X.ac3');

delete('front_right.wav','front_left.wav','front_center.wav','back_left.wav','back_right.wav','lfe.wav');

% Eliminem l'àudio de l'arxiu de vídeo i l'unim al binaural.
dd='Empezamos con el video'
s=length(fulpathname);
format=fulpathname(1,(s-3):s);
switch format
    case '.mkv'
        copyfile(fulpathname , [pwd '\orig.mkv']);
        system('ffmpeg -i orig.mkv -vcodec copy -an t.mkv ');
        system('ffmpeg -i audioBIN.wav -i t.mkv
videoOUT.mpg');
        delete('orig.mkv');
        delete('t.mkv');
    case '.mp4'
        copyfile(fulpathname , [pwd '\orig.mp4']);
        system('ffmpeg -i orig.mp4 -vcodec copy -an t.mp4 ');
        system('ffmpeg -i audioBIN.wav -i t.mp4
videoOUT.mpg');
        delete('orig.mp4');
        delete('t.mp4');
    case '.avi'
        copyfile(fulpathname , [pwd '\orig.avi']);
        system('ffmpeg -i orig.avi -vcodec copy -an t.avi ');
        system('ffmpeg -i audioBIN.wav -i t.avi
videoOUT.mpg');
        delete('orig.avi');
        delete('t.avi');
end

% Aquí mirem si l'usuari ha indicat on guardar l'arxiu final,
si no es guarda
% a l'arrel del programa
```

```
if(~isempty(dest_path))  
    movefile([pwd '\videoOUT.mpg'],dest_path,'f');  
end  
delete('audioBIN.wav');  
msgbox('Operation Completed');
```

## 7.2- "filtr\_HRTF.m"

```
function x = filtr_HRTF(audio , FS , angle_source , head_width)

% Aquesta funció filtra un audio MONO i en surt unde BINAURAL.
% Els parametres d'entrada són:
%   audio: Senyal d'àudio MONO
%   FS: Freqüència de mostreig del senyal "audio"
%   Angle_source: Àngle d'incidència de la font sonora.
%   head_width: Distància entre oïdes.
%   EXEMPLE:
%   x = filtr_HRTF(audio , 44100 , 50 , 17)

% Preparem un string amb l'angle que ens servirà per fer la cerca a la
BBDD
if (angle_source < 100 && angle_source >10)
    angle_source_string = ['0' num2str(angle_source)];
elseif (angle_source<10)
    angle_source_string = ['00' num2str(angle_source)];
else
    angle_source_string = num2str(angle_source);
end

if (angle_source==360)
    angle_source=0;
end

% 1º Llegir les HRTF L y R
pth=pwd;
pathname = [pth '\full\elev0\L0e' angle_source_string 'a.dat'];
fid = fopen(pathname,'r','ieee-be');
x1 = fread(fid,inf,'short');
fclose(fid);
% return as row vector, +/- 1 max.
x1 = x1' / 32768;

pathname = [pth '\full\elev0\R0e' angle_source_string 'a.dat'];
fid = fopen(pathname,'r','ieee-be');
```

```
xr = fread(fid,inf,'short');
fclose(fid);

% return as row vector, +/- 1 max.
xr = xr' / 32768;

% 2° Convolucionem en temps, per filtrar l'àudio en temps HRIR.
rightChannel=conv(audio,xr);
leftChannel=conv(audio,xl);

% 3° Calculem les mostres a afegir de delay entre L i R
%                               DelaySamples(    Lch ,
Rch    , angle_source , head_width , FS)
[leftChannelD , rightChannelD] = DelaySamples(leftChannel ,
rightChannel , angle_source , head_width , FS);
x= [leftChannelD ; rightChannelD]';%Unim els dos canals L i R en
una sola matriu que es pot llegir com un àudio estereo normal

end
```

## 7.3- "distancia LR.m"

```
function [L,R]=distancia_LR(ang,head_width,dist_source)

% Aquesta funció retorna la distància, en cm, entre una font sonora, a
% una distància
% 'dist_source', i cada una de les dues oïdes, L i R, a una distància
% head_width entre ambdues.
%     ang: en graus [0 , 360];
%     head_width: Distància interaural en cm;
%     dist_source: Distància en cm del centre del cap a la font.
%
%     EXAMPLE:
%     [L,R]=distancia_LR(120,17,1)

if (ang==360)
    ang=0;
end
if (ang<=90 && ang>=0)
    beta=90-ang;
    b=dist_source;
    c=head_width/2;
    R=sqrt((b.^2)+(c.^2)-2*b*c*cos(deg2rad(beta)));
    phi=180-beta;
    a=dist_source;
    L=sqrt((a.^2)+(c.^2)-2*a*c*cos(deg2rad(phi)));
elseif (ang>90&&ang<180)
    beta=((180-ang));
    b=dist_source;
    c=head_width/2;
    R=sqrt((b.^2)+(c.^2)-2*b*c*cos(deg2rad(beta)));
    phi=180-beta;
    a=dist_source;
    L=sqrt((a.^2)+(c.^2)-2*a*c*cos(deg2rad(phi)));
elseif (ang>=180&&ang<=270)
    beta=((270-ang));
    b=dist_source;
    c=head_width/2;
    L=sqrt((b.^2)+(c.^2)-2*b*c*cos(deg2rad(beta)));
```



```
    phi=180-beta;
    a=dist_source;
    R=sqrt((a.^2)+(c.^2)-2*a*c*cos(deg2rad(phi)));
elseif (ang>270&&ang<360)
    beta=((360-ang));
    b=dist_source;
    c=head_width/2;
    L=sqrt((b.^2)+(c.^2)-2*b*c*cos(deg2rad(beta)));
    phi=180-beta;
    a=dist_source;
    R=sqrt((a.^2)+(c.^2)-2*a*c*cos(deg2rad(phi)));
else
    error('Angulo debe estar entre 0 y 360');
end

end
```

## 7.4- "Delay Samples.m"

```
function [samplesL , samplesR] = DelaySamples(Lch , Rch , angle_source
, head_width , FS)

% Aquesta funció aplica un delay a un dels canals depenent de la
distància de la
% font a cada una de les oïdes, a l'oïda més llunyana a la font se li
aplica
% aquest retard mentre que a l'altra se li concatena un array de "0"
per què la
% mida dels arrays siguin iguals i és puguin concatenar.
%
% Com el que volem és la diferència de temps entre les dues oïdes, L i
R,
% suposarem que la font està a 1 m del centre del cap per simplificar
els
% càlculs.
%
% EXEMPLE:
% [samplesL , samplesR] = DelaySamples(Lch , Rch , 120 , 17 , 44100)

[L , R]=distancia_LR(angle_source,head_width,100); %Mirem la
distància de cada oïda a la font.

[NsL , NsR] = calcul_N_samples(L , R, FS);% Calculem el numero de
mostres per un temps determinat.

samplesLz = zeros(1,NsL(1));
samplesRz = zeros(1,NsR(1));

if(isempty(NsL))
    samplesL = [Lch' , samplesRz];
    samplesR = [samplesRz , Rch'];
else
    samplesL = [samplesLz , Lch'];
    samplesR = [Rch' , samplesLz];
end
end
```

## 7.5- "calcul\_N\_samples.m"

```
function [NsL , NsR] = calcul_N_samples(L , R , FS)

% La velocitat del so és la dinàmica de propagació de les ones
sonores.
% A l'atmosfera terrestre és de 343 m/s (a 20 °C de temperatura, amb
50
% d'humitat i a nivell de mar).

% Aquesta funció retorna el numero de mostres a afegir per aplicar a
% l'oïda més llunyana

%      L es la distància de l'oïda L a la font
%      R es la distància de l'oïda R a la font
%      FS la freqüència de mostreig

C = 343;

if(L>R) % la font es més proxima a R, hi ha que aplicar un delay
al audio L

%      dif/c és l'interval de temps que triga el so de més en
arribar a L
    d = (L-R)*0.01; %pasa de cm a metres
    NsL = round((d/C)*FS);
    NsR = 0;

elseif(R>L) % Aquest és el cas contrari
    d = (R-L)*0.01;
    NsR = round((d/C)*FS);
    NsL = 0;

else % L y R iguals

    NsR=0;
    NsL=0;
```

end  
end

## 7.6- "recerca.m"

```
%script de recerca.
%Inicializamos las tablas

tabla_m_hrtf_az15 = double.empty(187,0);
tabla_m_hrtf_az30 = double.empty(187,0);
tabla_m_hrtf_az45 = double.empty(187,0);
tabla_m_hrtf_az120 = double.empty(187,0);
tabla_m_hrtf_az135 = double.empty(187,0);
tabla_m_hrtf_az150 = double.empty(187,0);

for i=2:59
    if i<10
        user = ['0' num2str(i)];
    else
        user = num2str(i);
    end

    if(exist(['C:\Users\H\Desktop\PFG\Recerca\COMPENSATED\MAT\HRIR\IRC_10'
user '_C_HRIR.mat'])==2)
        individu =
load(['C:\Users\H\Desktop\PFG\Recerca\COMPENSATED\MAT\HRIR\IRC_10'
user '_C_HRIR.mat']);

        %Elevació 0°
        %Front:
        %
        EXAMPLE: m_oidioR_hrtf_azimut = abs(fft(hrir));

        m_oidoR_hrtf_az15 =
abs(fft(individu.r_eq_hrir_S.content_m(74,:)));
        m_oidoR_hrtf_az30 =
abs(fft(individu.r_eq_hrir_S.content_m(75,:)));
        m_oidoR_hrtf_az45 =
abs(fft(individu.r_eq_hrir_S.content_m(76,:)));
        %Rear:
        m_oidoR_hrtf_az120 =
abs(fft(individu.r_eq_hrir_S.content_m(81,:)));
```

```

        m_oidoR_hrtf_az135 =
abs(fft(individu.r_eq_hrir_S.content_m(82,:)));
        m_oidoR_hrir_az150 =
abs(fft(individu.r_eq_hrir_S.content_m(83,:)));

        tabla_m_hrtf_az15 = [tabla_m_hrtf_az15 ; m_oidoR_hrtf_az15];
        tabla_m_hrtf_az30 = [tabla_m_hrtf_az30 ; m_oidoR_hrtf_az30];
        tabla_m_hrtf_az45 = [tabla_m_hrtf_az45 ; m_oidoR_hrtf_az45];
        tabla_m_hrtf_az120 = [tabla_m_hrtf_az120 ;
m_oidoR_hrtf_az120];
        tabla_m_hrtf_az135 = [tabla_m_hrtf_az135 ;
m_oidoR_hrtf_az135];
        tabla_m_hrtf_az150 = [tabla_m_hrtf_az150 ;
m_oidoR_hrir_az150];

        end
end

%Carrquem les dades a un excel "tabla_m_hrtf_azX.xls"
xlswrite('tabla_m_hrtf_azX.xls',tabla_m_hrtf_az15,'15')
xlswrite('tabla_m_hrtf_azX.xls',tabla_m_hrtf_az30,'30')
xlswrite('tabla_m_hrtf_azX.xls',tabla_m_hrtf_az45,'45')
xlswrite('tabla_m_hrtf_azX.xls',tabla_m_hrtf_az120,'120')
xlswrite('tabla_m_hrtf_azX.xls',tabla_m_hrtf_az135,'135')
xlswrite('tabla_m_hrtf_azX.xls',tabla_m_hrtf_az150,'150')
clc
clear

```

## **7.7- "scrip analisis.m"**

```

% Fixem variables:
HW=17;
% Rear
angle_source_rear_right=150;
angle_source_rear_left=360-angle_source_rear_right;

```

```
% Front
angle_source_front_right=45;
angle_source_front_left=360-angle_source_front_right;

namewav='\audioBINaz0f45tr150.wav';

% Ruta a l'arxiu
[filename , pathname] =
uigetfile({'*.avi'; '*.mp4'; '*.mkv'; '*..*'}, 'Select the VIDEO file');
path=[pathname filename];

% Eliminar arxius residus possibles
delete('X.ac3');
delete('front_right.wav', 'front_left.wav', 'front_center.wav', 'back_left.wav', 'back_right.wav', 'lfe.wav');

% Extraiem el arxiu muticanal
ffmpegtranscode(path, 'x.ac3', 'Range' , [0 40] , 'Units' , 'seconds' ,
'FastSearch' , 'on' , 'VideoCodec' , 'none' );

% Desgranem canal a canal l'arxiu ac3
system('ffmpeg -i x.ac3 -filter_complex
channelsplit=channel_layout=5.1[FL][FR][FC][LFE][BL][BR] -map "[FL]"
front_left.wav -map "[FR]" front_right.wav -map "[FC]"
front_center.wav -map "[LFE]" lfe.wav -map "[BL]" back_left.wav -map
"[BR]" back_right.wav');

% Filtrem els diversos canals
%Llegim els 5 canals principals Front, 2xFronts & 2xRear
[x , FS] = audioread('back_left.wav'); %Llegir l'àudio
yrl = filtr_HRTF(x , FS , angle_source_rear_left , HW); %Filtrem
l'àudio
yrl = normaliza_audio_stereo(yrl); %Normalitzem l'àudio entre -1 i 1

[x , FS] = audioread('back_right.wav');
yrr = filtr_HRTF(x , FS , angle_source_rear_right , HW);
yrr = normaliza_audio_stereo(yrr);
```

```

d='rear done'

[x , FS] = audioread('front_center.wav');
yfc = filtr_HRTF(x , FS , 0 , HW);
yfc = normaliza_audio_stereo(yfc);

[x , FS] = audioread('front_left.wav');
yfl = filtr_HRTF(x , FS , angle_source_front_left , HW);
yfl = normaliza_audio_stereo(yfl);

[x , FS] = audioread('front_right.wav');
yfr = filtr_HRTF(x , FS , angle_source_front_right , HW);
yfr = normaliza_audio_stereo(yfr);
dd='front done'

[x , FS] = audioread('lfe.wav');
dd = 'se empieza el script de grabes'
% "greus" es una un pequeño script que crea una unica fuente de
% graves binaural como si viniesen los graves de los 5 puntos
% diferentes.
ylfe = greus(x , FS , HW , angle_source_front_left ,
angle_source_front_right , angle_source_rear_left ,
angle_source_rear_right);
dd = 'acaba el script de grabes'

% Suma d'audios
m = [length(yrl) , length(yfc) , length(yrr) , length(yfl) ,
length(yfr) , length(ylfe)];
m=min(m);
sum = yrl(1:m , :) + yfc(1:m , :) + yrr(1:m , :) + yfl(1:m , :) +
yfr(1:m , :) + ylfe(1:m , :);
sum = normaliza_audio_stereo(sum);%Normalització del àudio resultant

% Creem l'arxiu WAV final que unirem al vídeo.
% if(exist('audioBIN.wav')==2)
%     delete('audioBIN.wav');
% end
audiowrite([pwd namewav],sum,FS)

```



```
% Eliminar arxius residus possibles
delete('X.ac3');
delete('front_right.wav','front_left.wav','front_center.wav','back_left.wav','back_right.wav','lfe.wav');
clc;
clear;

msgbox('Operation Completed');
```